# Arbitrary Lagrangian-Eulerian-type discrete unified gas kinetic scheme for low-speed continuum and rarefied flow simulations with moving boundaries

Yong Wang (王勇),* Chengwen Zhong (钟诚文),† and Sha Liu (刘沙)‡
*National Key Laboratory of Science and Technology on Aerodynamic Design and Research,*
*Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China*

(Received 3 June 2019; revised manuscript received 10 November 2019; published 31 December 2019)

In this paper, the original discrete unified gas kinetic scheme (DUGKS) is extended to the arbitrary Lagrangian–Eulerian (ALE) framework to enable simulation of low-speed continuum and rarefied flows with moving boundaries. For the proposed ALE-type DUGKS, the mesh motion velocity is introduced in the Boltzmann–BGK equation and a remapping-free scheme is used to discretize the governing equation. Under this coupling framework, the complex rezoning and remapping phases implemented in the traditional ALE method are avoided. In some application areas, large discretization errors are introduced in the simulation if the geometric conservation law (GCL) is not guaranteed. Therefore, three GCL-compliant approaches are discussed, and a uniform flow test case is conducted to validate these schemes. Further, to illustrate the performance of the proposed method, four test cases are simulated, including the continuum flow around an oscillating circular cylinder, the continuum flow around a pitching NACA0012 airfoil, a moving piston driven by a rarefied gas, and the rarefied flow caused by a plate oscillating in the normal direction. Finally, an extended test case considering the rarefied flow over an oscillating circular cylinder is also studied, as this condition is not sufficiently researched. Consistent and good results obtained from the above test cases demonstrate the capability of the proposed ALE-type DUGKS to simulate moving boundary problems in different flow regimes.

DOI: 10.1103/PhysRevE.100.063310

## I. INTRODUCTION

Moving boundary problems can be found in various scientific and engineering fields; for example, in the field of aerospace engineering, the separation of the store from an aircraft body and the motion of the undercarriage during take-off and landing [1]. For microair vehicles, the design of rotary and flapping wings, which include a large number of moving boundary problems, is another application domain [2]. In the above examples, usually the reference lengths of flows are considerably larger than the mean free path of gas molecules. Following the definition of the Knudsen number [3], these flows are in a continuum flow regime. Similarly, moving boundary problems are also encountered in the application of a rarefied flow regime, such as the sound wave generated by an oscillating plate and the motion of vanes of a Crookes radiometer [4]. In general, macromethods based on Navier–Stokes equations [5] can simulate moving boundary problems in a continuum flow regime, and the prevailing direct simulation Monte Carlo (DSMC) method [6] can deal with the problem in a rarefied flow regime. However, in some applications, such as flows involving multiple flow regimes, both macromethods and DSMC will fail because the flow regimes are outside the scope of these methods. Although some hybrid methods [2,7] have been developed, they also face great difficulties due to different temporal and spatial scales. Therefore, developing and/or improved a method that can simulate moving boundary problems in all flow regimes will have a great value in engineering applications.

Recently, the discrete unified gas kinetic scheme (DUGKS) proposed by Guo *et al.* [8] has become a new promising method to simulate flows in all regimes. This method combines the advantages of the lattice Boltzmann method [9] (LBM) and the unified gas kinetic scheme [10] (UGKS), which enables easy calculation of the flux at a cell interface similar to the LBM and reduces the computational cost compared to that of the UGKS. Some details can be found in Refs. [8,11–14]. Currently, the DUGKS is implemented on a stationary mesh. Therefore, this study aims to present an improved DUGKS that can simulate moving boundary problems in all flow regimes.

Based on the mesh systems used in the numerical computation, the existing methods to simulate moving boundary problems can be divided into two categories: the Eulerian method and Lagrangian method. For the Eulerian method, the mesh is fixed at each iteration computational step. The immersed boundary method (IBM) can be considered as an example of the Eulerian method [15–17]. In this method, uniform Cartesian grid cells are used near the wall region, and the moving boundary is discretized into a set of Lagrangian points. To implement the wall boundary condition, some interpolation methods between the Eulerian points (Cartesian grid points) and the Lagrangian points are used to guarantee the no-slip condition. Except for the applications in continuum flow regime, the IBM coupled with the UGKS can also deal with moving boundary problems in a rarefied flow regime [18]. The primary disadvantage of the IBM is that in some applications, such as in flows with a high Reynolds

*wongyung@mail.nwpu.edu.cn
†zhongcw@nwpu.edu.cn
‡shaliu@nwpu.edu.cn

number, the number of grid cells is extremely large. The static mesh movement method [19] is another representation of the Eulerian method. During the numerical simulation, after an Eulerian step, a new mesh is regenerated based on some criteria. In general, the tasks of regenerating a new mesh and implementing some interpolation methods to transfer flow variables from old to new meshes are time consuming for most applications. In the Lagrangian method, the mesh motion velocity is equal to the local fluid flow velocity, which often makes mesh distortion and tangling unavoidably. Consequently, the pure Lagrangian method usually performs well for one-dimensional flow simulations, but face great difficulties for two- or three-dimensional flow simulations in most applications [20].

By combining the advantages of the Eulerian and Lagrangian methods, a famous technique called the arbitrary Lagrangian–Eulerian [21] (ALE) method has been developed and improved during the last few decades. The ALE method can also be divided into two types. In the traditional procedure, three steps, namely, the explicit Lagrangian phase, the rezoning phase, and the remapping phase, are implemented. Similar to the static mesh movement method developed in the pure Eulerian method, mesh regeneration or modification and flow variable transfer are two critical steps in this type of ALE method. That is, if the mesh quality can be maintained well during the moving process, then this ALE-type method becomes a pure Lagrangian method. In another type of ALE method, usually called the remapping-free ALE, the mesh motion velocity is introduced in the convection terms of governing equations to modify the net flux of a cell interface. In addition, a mesh moving technique is also introduced, and the mesh motion velocity is constructed on the basis of the old and new meshes. In the aerospace field, such as in aeroelastic analysis, this ALE-type method is usually used. In general, the time required to implement a mesh moving technique is less than that to regenerate a new mesh; moreover, the rezoning and remapping phases can also be eliminated. Therefore, in this study, the remapping-free ALE technique is used to improve the original DUGKS.

In the simulation of moving boundary problems, one source of numerical error is the violation of the geometric conservation law (GCL), which, in some applications, even leads to erroneous results. Following Chang *et al.* [22], in this study, GCL-compliant schemes are considered in order to exclude this numerical error.

The rest of the paper is organized as follows. In Sec. II, the original DUGKS is introduced briefly, and the ALE-type DUGKS and several GCL schemes are illustrated in detail. In Sec. III, a test case is conducted to verify the GCL schemes, and five test cases are conducted to validate the capability of the present method in simulating the moving boundary problems. Finally, a brief conclusion is presented in Sec. IV.

## II. NUMERICAL METHODS

### A. Sketch of the original discrete unified gas kinetic scheme

In this section, the original DUGKS proposed by Guo *et al.* [8] is introduced briefly. The starting point of the DUGKS is the Boltzmann–BGK equation, which can be expressed as

$$\frac{\partial f}{\partial t} + \boldsymbol{\xi} \cdot \nabla f = \Omega = -\frac{1}{\tau}[f - f^{\mathrm{eq}}], \tag{1}$$

where $f = f(\boldsymbol{x}, \boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\zeta}, t)$ is the velocity distribution function for particles moving in a $D$-dimensional velocity space with $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_D)$ at position $\boldsymbol{x} = (x_1, \ldots, x_D)$ and time $t$. $\boldsymbol{\eta} = (\xi_{D+1}, \ldots, \xi_3)$ is the rest components of the particle velocity with length $L = 3 - D$. $\boldsymbol{\zeta}$ is a vector with $K$ dimension, representing the internal degree of freedom of molecules. $\tau$ is the relaxation time relating to the fluid dynamics viscosity $\mu$ and pressure $p$ with $\tau = \mu/p$. Finally, $f^{\mathrm{eq}}$ is the Maxwellian equilibrium distribution function given by

$$f^{\mathrm{eq}} = \frac{\rho}{(2\pi RT)^{(3+K)/2}} \exp\left(-\frac{c^2 + \eta^2 + \zeta^2}{2RT}\right), \tag{2}$$

where $R$ is the gas constant, $T$ is the fluid temperature, $\rho$ is the fluid density, and $\boldsymbol{c} = (\boldsymbol{\xi} - \boldsymbol{u})$ is the peculiar velocity with $\boldsymbol{u}$ being the macroscopic flow velocity.

To remove the dependence of a distribution function on the internal variable $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$, usually the following two reduced distributions [23], namely, the density distribution function $g$ and the energy distribution function $h$, are introduced into practical computation:

$$g(\boldsymbol{x}, \boldsymbol{\xi}, t) = \int f(\boldsymbol{x}, \boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\zeta}, t) d\boldsymbol{\eta} d\boldsymbol{\zeta}, \tag{3}$$

$$h(\boldsymbol{x}, \boldsymbol{\xi}, t) = \int (\eta^2 + \zeta^2) f(\boldsymbol{x}, \boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\zeta}, t) d\boldsymbol{\eta} d\boldsymbol{\zeta}. \tag{4}$$

Then, with Eq. (1), the evolution equations for $g$ and $h$ can be expressed as

$$\frac{\partial g}{\partial t} + \boldsymbol{\xi} \cdot \nabla g = \Omega_g = -\frac{1}{\tau}[g - g^{\mathrm{eq}}], \tag{5}$$

$$\frac{\partial h}{\partial t} + \boldsymbol{\xi} \cdot \nabla h = \Omega_h = -\frac{1}{\tau}[h - h^{\mathrm{eq}}], \tag{6}$$

respectively, where the equilibrium distribution functions for $g^{\mathrm{eq}}$ and $h^{\mathrm{eq}}$ are given by

$$g^{\mathrm{eq}} = \int f^{\mathrm{eq}} d\boldsymbol{\eta} d\boldsymbol{\zeta} = \frac{\rho}{(2\pi RT)^{D/2}} \exp\left(-\frac{c^2}{2RT}\right), \tag{7}$$

$$h^{\mathrm{eq}} = (K + 3 - D) RT g^{\mathrm{eq}}. \tag{8}$$

The macrophysical quantities can be calculated as

$$\rho = \int g d\boldsymbol{\xi}, \quad \rho \boldsymbol{u} = \int \boldsymbol{\xi} g d\boldsymbol{\xi}, \quad \rho E = \frac{1}{2} \int (\xi^2 g + h) d\boldsymbol{\xi}, \tag{9}$$

and with the ideal gas law, $p = \rho RT$, the pressure can also be obtained. In addition, the relationship between dynamic viscosity $\mu$ and temperature $T$, based on the hard-sphere (HS) or variable hard-sphere (VHS) molecules, is given by

$$\mu = \mu_{\mathrm{ref}} \left(\frac{T}{T_{\mathrm{ref}}}\right)^{\omega}, \tag{10}$$

where $\omega$ is the index related to the HS or VHS model, and $\mu_{\mathrm{ref}}$ is the fluid viscosity at the reference temperature $T_{\mathrm{ref}}$.

As Eqs. (5) and (6) are exactly the same, a uniform expression can be used:

$$\frac{\partial \phi}{\partial t} + \boldsymbol{\xi} \cdot \nabla \phi = \Omega = -\frac{1}{\tau}[\phi - \phi^{\mathrm{eq}}], \tag{11}$$

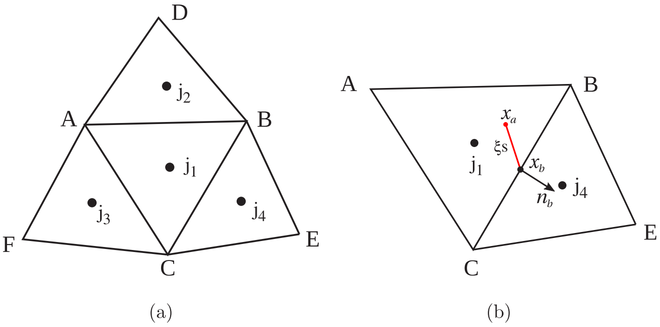


FIG. 1. Schematics of (a) an unstructured mesh used in the DUGKS and (b) flux calculation for a cell interface.

where $\phi$ represents $g$ or $h$. In the DUGKS, Eq. (11) is solved using a finite volume method. The discretization of this equation can be divided into two steps: velocity-space discretization and physical-space discretization.

For the particle velocity-space discretization, usually, a finite set of discretized microvelocities is used [8]; $\boldsymbol{\xi}_i$ represents the $i$th discretized velocity. As the update of macroquantity depends on the particle microvelocities [Eq. (9)], the values of these velocities can be set to coincide with the abscissas of the quadrature rule. For low-speed continuum flows, the discretized velocities, weights, and the corresponding equilibrium distribution functions developed in the LBM, such as the D2Q9 model [24], can be used for the DUGKS. In addition, this approach builds a connection between the LBM and DUGKS. For rarefied flows, usually, the Gauss–Hermit and Newton–Cotes quadrature rules are used to integrate the macroquantities, and the corresponding set of abscissas is used as the set of discretized velocities.

For the physical-space discretization, in this study, a finite volume method based on an unstructured mesh is used. Figure 1(a) shows the schematic of an unstructured mesh; $j$ is the center of a triangle cell $ABC$, and subscript represents the index number of a cell. If $\phi_j$ and $\Omega_j$ are the average values of $\phi$ and $\Omega$, respectively, in cell $ABC$, $\Delta t = t^{n+1} - t^n$ is the time step, and the midpoint rule for the time integration of the convection term and the trapezoidal rule for the collision term are used, then Eq. (11) can be rewritten as

$$\phi_j^{n+1}(\boldsymbol{\xi}) - \phi_j^n(\boldsymbol{\xi}) + \frac{\Delta t}{|V_j|}\boldsymbol{F}^{n+1/2}(\boldsymbol{\xi}) = \frac{\Delta t}{2}\left[\Omega_j^{n+1}(\boldsymbol{\xi}) + \Omega_j^n(\boldsymbol{\xi})\right], \tag{12}$$

where $V_j$ is the volume of cell $ABC$. $\boldsymbol{F}^{n+1/2}(\boldsymbol{\xi})$ is the flux of the cell surface, which is given by

$$\boldsymbol{F}^{n+1/2}(\boldsymbol{\xi}) = \int_{\partial V_j} (\boldsymbol{\xi}\cdot\boldsymbol{n})\phi(\boldsymbol{x}, \boldsymbol{\xi}, t_{n+1/2})dS, \tag{13}$$

where $\partial V_j$ represents the cell surface, $\boldsymbol{x}$ is the center of the cell interface, and $\boldsymbol{n}$ is the outward unit vector normal to the surface. To remove the implicit collision term, the following two new distribution functions are introduced:

$$\tilde{\phi} = \phi - \frac{\Delta t}{2}\Omega = \frac{2\tau + \Delta t}{2\tau}\phi - \frac{\Delta t}{2\tau}\phi^{\mathrm{eq}}, \tag{14}$$

$$\tilde{\phi}^+ = \phi + \frac{\Delta t}{2}\Omega = \frac{2\tau - \Delta t}{2\tau + \Delta t}\tilde{\phi} + \frac{2\Delta t}{2\tau + \Delta t}\phi^{\mathrm{eq}}. \tag{15}$$

Then, Eq. (12) can be rewritten as

$$\tilde{\phi}_j^{n+1} = \tilde{\phi}_j^{+,n} - \frac{\Delta t}{|V_j|}\boldsymbol{F}^{n+1/2}(\boldsymbol{\xi}). \tag{16}$$

With the conservative property of the collision term, in practical computation, $\tilde{\phi}$ is solved instead of $\phi$, and Eq. (9) is rewritten as

$$\rho = \int \tilde{g}d\boldsymbol{\xi}, \quad \rho\boldsymbol{u} = \int \boldsymbol{\xi}\tilde{g}d\boldsymbol{\xi}, \quad \rho E = \frac{1}{2}\int (\xi^2\tilde{g} + \tilde{h})d\boldsymbol{\xi}. \tag{17}$$

For the calculation of the interface flux shown in Fig. 1(b), $\boldsymbol{x}_b$ is the center of interface $BC$ and $\boldsymbol{n}_b$ is the unit normal vector of $BC$ with the direction points from cell $ABC$ to $BEC$. If Eq. (11) is integrated along the characteristic line within a half time step $s = \Delta t/2$, then the original distribution function $\phi(\boldsymbol{x}_b, \boldsymbol{\xi}, t_n + s)$ in Eq. (13) can be updated as

$$\begin{aligned}\phi(\boldsymbol{x}_b, \boldsymbol{\xi}, t_n + s) &- \phi(\boldsymbol{x}_b - \boldsymbol{\xi}s, \boldsymbol{\xi}, t_n)\\ &= \frac{s}{2}[\Omega(\boldsymbol{x}_b, \boldsymbol{\xi}, t_n + s) + \Omega(\boldsymbol{x}_b - \boldsymbol{\xi}s, \boldsymbol{\xi}, t_n)].\end{aligned} \tag{18}$$

Similar to the treatment of $\tilde{\phi}$, the following two new distribution functions are introduced:

$$\bar{\phi} = \phi - \frac{s}{2}\Omega = \frac{2\tau + s}{2\tau}\phi - \frac{s}{2\tau}\phi^{\mathrm{eq}}, \tag{19}$$

$$\bar{\phi}^+ = \phi + \frac{s}{2}\Omega = \frac{2\tau - s}{2\tau + s}\bar{\phi} + \frac{2s}{2\tau + s}\phi^{\mathrm{eq}}. \tag{20}$$

Then, Eq. (18) can be rewritten as

$$\bar{\phi}(\boldsymbol{x}_b, \boldsymbol{\xi}, t_{n+1/2}) = \bar{\phi}^+(\boldsymbol{x}_b - \boldsymbol{\xi}s, \boldsymbol{\xi}, t_n). \tag{21}$$

If $\tilde{\phi}$ in Eq. (17) is replaced by $\bar{\phi}$, then the macroquantities at a cell interface can also be calculated. Further, with Eq. (19), the original distribution function at $(\boldsymbol{x}_b, t_{n+1/2})$ is given as

$$\begin{aligned}\phi(\boldsymbol{x}_b, \boldsymbol{\xi}, t_{n+1/2}) = &\frac{2\tau}{2\tau + s}\bar{\phi}(\boldsymbol{x}_b, \boldsymbol{\xi}, t_n + s)\\ &+ \frac{s}{2\tau + s}\phi^{\mathrm{eq}}(\boldsymbol{x}_b, \boldsymbol{\xi}, t_n + s).\end{aligned} \tag{22}$$

Furthermore, as illustrated by Eq. (21) and Fig. 1(b), $\phi^{n+1/2}$ at a cell interface is obtained with $\bar{\phi}^+$ at $\boldsymbol{x}_a = \boldsymbol{x}_b - \boldsymbol{\xi}s$. In addition, with Eqs. (14), (19), and (20), the relationship between $\tilde{\phi}$ and $\bar{\phi}^+$ is built as follows:

$$\bar{\phi}^+ = \frac{2\tau - s}{2\tau + \Delta t}\tilde{\phi} + \frac{3s}{2\tau + \Delta t}\phi^{\mathrm{eq}}. \tag{23}$$

Hence, with $\tilde{\phi}$ and $\phi^{\mathrm{eq}}$ stored at $\boldsymbol{x}_j$ and with the Taylor expansion, $\phi^+(\boldsymbol{x}_a, t_n)$ is calculated as

$$\bar{\phi}^+(\boldsymbol{x}_a, t_n) = \bar{\phi}^+(\boldsymbol{x}_j, t_n) + (\boldsymbol{x}_a - \boldsymbol{x}_j)\cdot\nabla\bar{\phi}^+(\boldsymbol{x}_j, t_n), \tag{24}$$

where $\nabla\bar{\phi}^+$ is the gradient of $\bar{\phi}^+$. For the calculation of $\nabla\bar{\phi}^+$ on an unstructured mesh, in this study, a least-squares method is used:

$$\min_{\nabla\bar{\phi}_j^+} \sum_m w_{j,m}[\bar{\phi}_{j,m}^+ - \bar{\phi}_j^+ - \nabla\bar{\phi}_j^+\cdot(\boldsymbol{x}_{j,m} - \boldsymbol{x}_j)]^2, \tag{25}$$

where $w_{j,m} = 1/(\boldsymbol{x}_{j,m} - \boldsymbol{x}_j)^2$ is the geometrical weighting factor and $m$ is the total number of cell neighbors. In addition, in practical computation, after $\bar{\phi}^+$ is calculated, with Eqs. (15) and (23), $\tilde{\phi}^+$ in Eq. (16) is updated as

$$\tilde{\phi}^+ = \tfrac{4}{3}\bar{\phi}^+ - \tfrac{1}{3}\tilde{\phi}. \tag{26}$$

### B. ALE-type discrete unified gas kinetic scheme

#### *1. Discretization of the ALE-type DUGKS*

In this section, discretization of the ALE-type DUGKS will be introduced in detail. Under the ALE framework, during the simulation, the geometrical information of a grid cell, such as volume of cell, the location of the cell center, and the length (or area) of the cell interface, changes temporally. Following the macromethod based on N–S equations and the UGKS [25] based on the Boltzmann–BGK equation, a mesh motion velocity $\boldsymbol{v}$ that modifies the net flux of a cell interface is introduced, and Eq. (1) is rewritten as

$$\frac{\partial f}{\partial t} + (\boldsymbol{\xi} - \boldsymbol{v}) \cdot \nabla f = \Omega = -\frac{1}{\tau}[f - f^{\text{eq}}]. \tag{27}$$

Note that in Eq. (27), the calculation of the equilibrium distribution function $f^{\text{eq}}$ does not depend on the mesh motion velocity $\boldsymbol{v}$ [26], and the macroquantities stored in the global coordinate are used to update $f^{\text{eq}}$. In other words, velocity $\boldsymbol{v}$ only influences the calculation of the convection term in the Boltzmann–BGK equation. The reason presented in Ref. [26] is that during the coordinate transformation from the global (physical domain) to the local (computational domain) coordinates, both the microparticle velocity and the macrofluid velocity will include the same expression of velocity $\boldsymbol{v}$ (Eqs. (7) and (11) in Ref. [26]), so the influence of velocity $\boldsymbol{v}$ to $f^{\text{eq}}$ disappears. Although no coordinate transformation is required in DUGKS for the calculation of the interface flux, the idea presented in Ref. [26] can also explain the confusion in the calculation of $f^{\text{eq}}$.

In the ALE-type DUGKS, by using the discretization scheme presented in the original DUGKS, Eqs. (12) and (13) will be rewritten as

$$\begin{aligned}\phi_j^{n+1}(\boldsymbol{\xi})\left|V_j^{n+1,*}\right| - \phi_j^{n}(\boldsymbol{\xi})\left|V_j^{n,*}\right| + \Delta t \boldsymbol{F}_{\text{ALE}}^{n+1/2}(\boldsymbol{\xi})\\ = \frac{\Delta t}{2}\left[\Omega_j^{n+1}(\boldsymbol{\xi})\left|V_j^{n+1,*}\right| + \Omega_j^{n}(\boldsymbol{\xi})\left|V_j^{n,*}\right|\right],\end{aligned} \tag{28}$$

and

$$\begin{aligned}\boldsymbol{F}_{\text{ALE}}^{n+1/2}(\boldsymbol{\xi}) &= \int_{\partial V_j} (\boldsymbol{\xi} - \boldsymbol{v}) \cdot \boldsymbol{n}\phi(\boldsymbol{x}, \boldsymbol{\xi}, t_{n+1/2})dS\\ &= \sum_k \left(\boldsymbol{\xi} - \boldsymbol{v}_{b,k}^{n+1/2}\right) \cdot \boldsymbol{n}_{b,k}^{*}\phi^{n+1/2}(\boldsymbol{x}_{b,k}, \boldsymbol{\xi})S_k^{*},\end{aligned} \tag{29}$$

respectively, where $V^{n+1,*}$ and $V^{n,*}$ are the cell volumes at $n+1$ and $n$ time levels; superscript $^*$ means that the values of volume at corresponding times maybe not equal to the real values of volume at those times and will be illustrated in the next section. $k$ is the total number of cell interfaces, $\boldsymbol{v}_b^{n+1/2}$ is the motion velocity of the cell interface at $n + 1/2$ time, and $\boldsymbol{n}_b^*$ and $S_b^*$ are the outward unit normal vector and the area of the cell interface, respectively. The computational method for these three variables will also be illustrated in the subsequent section.

Similar to the original DUGKS, in the ALE-type DUGKS, two new distribution functions, $\tilde{\phi}$ and $\tilde{\phi}^+$, are introduced; Eq. (28) can be rewritten as

$$\begin{aligned}\tilde{\phi}_j^{n+1} &= \left|\frac{V_j^{n,*}}{V_j^{n+1,*}}\right|\tilde{\phi}_j^{+,n} - \frac{\Delta t}{\left|V_j^{n+1,*}\right|}\boldsymbol{F}_{\text{ALE}}^{n+1/2}(\boldsymbol{\xi})\\ &= |V_{j,\text{mod}}|\tilde{\phi}_j^{+,n} - \frac{\Delta t}{\left|V_j^{n+1,*}\right|}\boldsymbol{F}_{\text{ALE}}^{n+1/2}(\boldsymbol{\xi}),\end{aligned} \tag{30}$$

where the expressions of $\tilde{\phi}_j^{n+1}$ and $\tilde{\phi}_j^{+,n}$ are the same as those in the original DUGKS. The method for reconstructing $\phi^{n+1/2}$ at the cell interface is also the same as that in the original DUGKS, where $\boldsymbol{x}_a = \boldsymbol{x}_b^n - \boldsymbol{\xi}s$ is used to calculate the location of interpolating point. In addition, the sign of $\boldsymbol{\xi}\boldsymbol{n}_b^n$ is used to judge the upwind direction. Finally, from the perspective of programming, the following three modifications are made to the framework of the original DUGKS solver in order to form an ALE-type DUGKS solver, which can simulate moving boundary problems in both continuum and rarefied flow regimes:

(1) addition of an independent module for mesh motion and updating of geometrical information into the old solver,

(2) introduction of the volume-modified coefficient $|V_{\text{mod}}|$ into Eq. (16),

(3) replacement of the vector product $\boldsymbol{\xi} \cdot \boldsymbol{n}$ in Eq. (13) by $(\boldsymbol{\xi} - \boldsymbol{v}) \cdot \boldsymbol{n}$.

In addition, in this paper, the Laplace smoothing equation for mesh deformation [27] is solved to update the unstructured mesh.

#### *2. Geometric conservation law*

The concept of the GCL was first defined by Thomas and Lambard in 1978 [28]. In general, for a uniform flow, if a scheme based on the moving mesh is GCL compliant, any disturbance must not be introduced into the flow domain at any time. Therefore, the 'free-stream preservation property' is a fundamental condition for any ALE schemes [22]. Similar to the theoretical derivation used in the macromethod, the Boltzmann–BGK equation is considered here. By integrating Eq. (27) on a control volume, we have

$$\frac{d}{dt}\int f dV + \int_{\partial V} (\boldsymbol{\xi} - \boldsymbol{v}_b) \cdot \boldsymbol{n}_b f_b dS = -\frac{1}{\tau}\int [f - f^{\text{eq}}]dV. \tag{31}$$

Then, for a uniform flow, that is, the distribution function $f$ in each cell maintains the original Maxwell equilibrium state at any time ($f = f^{\text{eq}}$), and with $\int \boldsymbol{n}_b dS = 0$, based on the semidiscrete scheme, Eq. (31) can be simplified as (here we assume that $\boldsymbol{\xi}$ is discretized by the method described in Sec. II A)

$$\frac{d}{dt}V_j = \int_{\partial V} \boldsymbol{v}_b \cdot \boldsymbol{n}_b dS = \sum_k \boldsymbol{v}_{b,k}\boldsymbol{n}_{b,k}S_{b,k} = \sum_k \boldsymbol{v}_{b,k}\boldsymbol{S}_{b,k}. \tag{32}$$

Equation (32) is the governing equation of the GCL and it implies that the volume variation of a moving grid cell equals the integration of the volume flux (or "sweeping volume") of

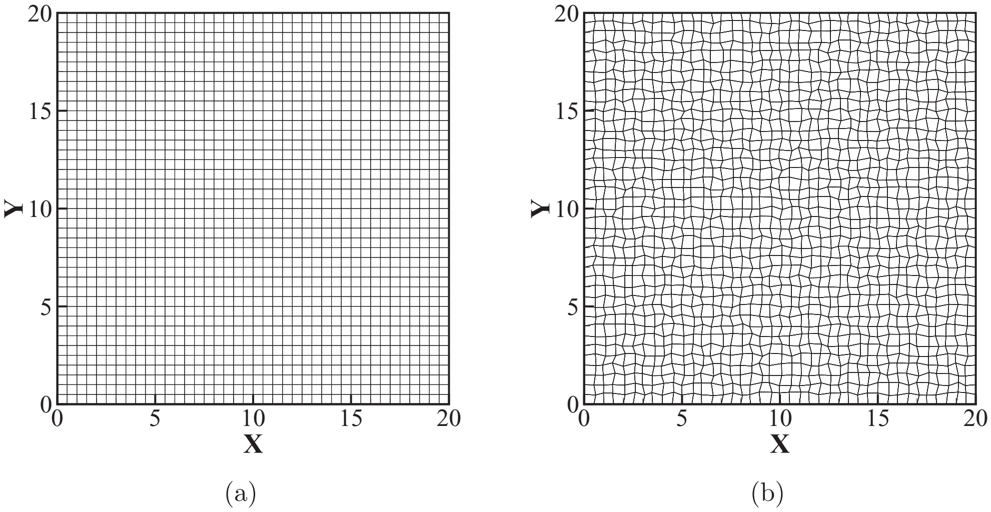


FIG. 2. Mesh for a uniform flow: (a) initial mesh and (b) instantaneous mesh.

all the moving cell interfaces surrounding the control volume [22].

In this study, the motion velocity $\boldsymbol{v}_b$ of a cell interface in Eq. (29) is given as

$$\boldsymbol{v}_b^{n+1/2} = \frac{\boldsymbol{x}_b^{n+1} - \boldsymbol{x}_b^n}{\Delta t}. \tag{33}$$

Based on the mesh motion velocity $\boldsymbol{v}$ and the GCL governing equation, three discretized geometric conservation law (DGCL) compliant schemes to determine the values of $V_j^{n+1,*}$, $V_j^{n,*}$, and $\boldsymbol{S}_b^*$ are presented.

(1) DGCL scheme 1:

If we set

$$V_j^{n+1,*} = V_j^{n+1}, V_j^{n,*} = V_j^n, \tag{34}$$

which implies that real values of volume at corresponding times are used, then following the idea presented in Ref. [29], $\boldsymbol{S}_b^*$ can be calculated as

$$\boldsymbol{S}_b^* = \boldsymbol{S}_b^{n+\frac{1}{2}} = \frac{\boldsymbol{S}_b^{n+1} + \boldsymbol{S}_b^n}{2}. \tag{35}$$

With a simple mathematical derivation, Eqs. (34) and (35) will automatically satisfy Eq. (32) during the mesh motion process.

(2) DGCL scheme 2:

If we set

$$V_j^{n,*} = V_j^n, \boldsymbol{S}_b^* = \boldsymbol{S}_b^n, \tag{36}$$

to satisfy the DGCL, then $V^{n+1,*}$ must be modified [22]. With the first-order Euler temporal discretization scheme, Eq. (32) can be rewritten as

$$\frac{V_j^{n+1} - V_j^n}{\Delta t} = \sum_k \boldsymbol{v}_{b,k}^{n+1/2} \boldsymbol{S}_{b,k}^n, \tag{37}$$

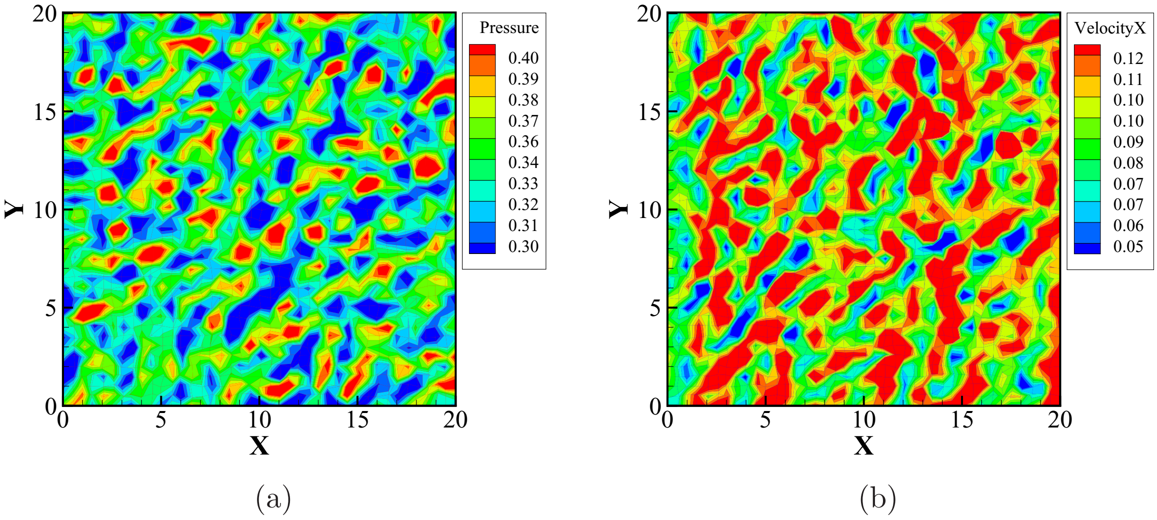


FIG. 3. (a) Pressure and (b) velocity contours of uniform flow obtained with a DGCL-violated scheme.

TABLE I. Spatial errors of numerical solutions for a uniform flow.

| | | Pressure | | $U$ | | $V$ | |
|---|---|---|---|---|---|---|---|
| $\Delta x$ (cell size) | DGCL scheme | $L_2$ | $L_\infty$ | $L_2$ | $L_\infty$ | $L_2$ | $L_\infty$ |
| 1.000 | Scheme 1 | 7.44*e*-14 | 1.17*e*-14 | 2.29*e*-13 | 4.07*e*-14 | 3.38*e*-13 | 6.02*e*-14 |
| | Scheme 2 | 1.15*e*-13 | 2.14*e*-14 | 5.98*e*-13 | 1.19*e*-13 | 7.47*e*-13 | 1.21*e*-13 |
| | Scheme 3 | 8.07*e*-14 | 5.88*e*-15 | 1.21*e*-13 | 2.04*e*-14 | 9.86*e*-14 | 2.99*e*-14 |
| 0.500 | Scheme 1 | 3.00*e*-13 | 1.52*e*-14 | 2.95*e*-13 | 3.48*e*-14 | 2.56*e*-13 | 2.09*e*-14 |
| | Scheme 2 | 2.63*e*-13 | 9.55*e*-15 | 2.81*e*-13 | 3.29*e*-14 | 2.25*e*-13 | 1.85*e*-14 |
| | Scheme 3 | 2.95*e*-13 | 1.02*e*-14 | 2.27*e*-13 | 2.15*e*-14 | 1.90*e*-13 | 1.31*e*-14 |
| 0.250 | Scheme 1 | 1.10*e*-12 | 4.10*e*-14 | 7.55*e*-13 | 3.84*e*-14 | 6.79*e*-13 | 2.67*e*-14 |
| | Scheme 2 | 1.04*e*-12 | 1.64*e*-14 | 6.48*e*-13 | 2.96*e*-14 | 5.73*e*-13 | 1.80*e*-14 |
| | Scheme 3 | 9.63*e*-13 | 1.52*e*-14 | 6.23*e*-13 | 2.56*e*-14 | 5.48*e*-13 | 1.51*e*-14 |
| 0.125 | Scheme 1 | 3.33*e*-12 | 5.71*e*-14 | 1.96*e*-12 | 4.38*e*-14 | 1.91*e*-12 | 4.45*e*-14 |
| | Scheme 2 | 2.64*e*-12 | 2.10*e*-14 | 1.45*e*-12 | 2.02*e*-14 | 1.47*e*-12 | 1.76*e*-14 |
| | Scheme 3 | 2.73*e*-12 | 2.14*e*-14 | 1.48*e*-12 | 1.94*e*-14 | 1.50*e*-12 | 1.82*e*-14 |

and $V_j^{n+1,*}$ can be modified as

$$V_j^{n+1,*} = V_j^n + \Delta t \sum_k \boldsymbol{v}_{b,k}^{n+1/2} \boldsymbol{S}_{b,k}^n. \tag{38}$$

(3) DGCL scheme 3:
If we set

$$V_j^{n+1,*} = V_j^{n+1}, \boldsymbol{S}_b^* = \boldsymbol{S}_b^{n+1}, \tag{39}$$

similar to DGCL scheme 2, then $V_j^{n,*}$ also must be modified and it can be calculated as

$$V_j^{n,*} = V_j^{n+1} - \Delta t \sum_k \boldsymbol{v}_{b,k}^{n+1/2} \boldsymbol{S}_{b,k}^{n+1}. \tag{40}$$

Since DGCL scheme 1 is considerably easy to implement and only one time level [$\boldsymbol{F}^{n+1/2}$ in Eq. (29)] is needed to calculate the flux at a cell interface, it can naturally be coupled with the current ALE-type DUGKS. Although DGCL schemes 2 and 3 are relatively more complex, they are suitable for a further improved ALE-type DUGKS framework, such as a high-order DUGKS [30] or a multi-time-level implicit scheme [22], where the geometrical information at those multi-time-levels is considerably more difficult to define [31]. In addition, these two schemes are volume constrained; a face-constrained scheme [22] can also be used, but it is not considered in this study.

### *3. Boundary conditions*

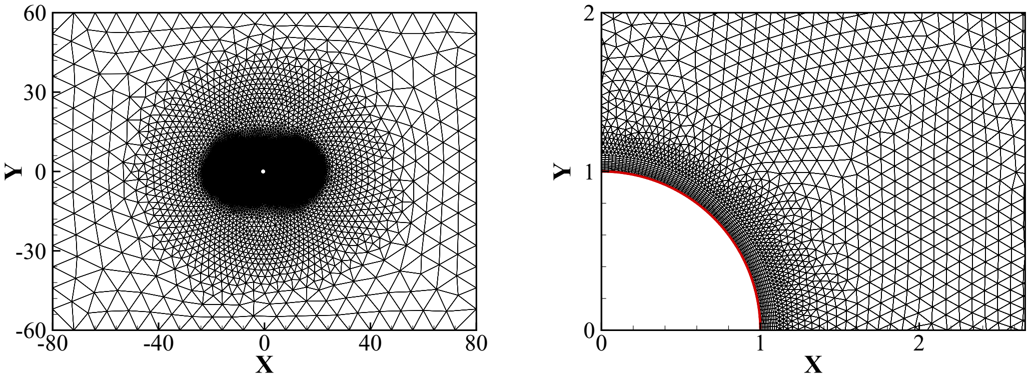
In this section, the boundary conditions used in this study will be illustrated in detail.

For the wall boundary condition, depending on whether the flow condition is continuum or rarefied, the non-equilibrium extrapolation rule [32] or diffuse-scattering rule [8] will be used. For a continuum flow, the original distribution functions at $n + 1/2$ are given as

$$f_w^{n+1/2}(\boldsymbol{\xi}_i) = f_w^{\text{eq}}(\boldsymbol{\xi}_i; \rho_w, \boldsymbol{u}_w) + f_j^{n+1/2}(\boldsymbol{\xi}_i) - f_j^{\text{eq},n+1/2}(\boldsymbol{\xi}_i), \tag{41}$$

where subscript $w$ represents the wall boundary, $j$ is the neighbor cell of a wall interface, and $\rho_w$ and $\boldsymbol{u}_w$ are the density and velocity at the wall, respectively. In addition, $\rho_w$ can be approximated by $\rho_w = \rho_j$, and $f_j^{n+1/2}$ and $f_j^{\text{eq},n+1/2}$ can be approximated by the corresponding values at $n$ time. For a rarefied flow, the distribution function for particles moving in the direction reflecting from the wall is calculated as

$$f_w^{n+1/2}(\boldsymbol{\xi}_i) = f^{\text{eq}}(\boldsymbol{\xi}_i; \rho_w, \boldsymbol{u}_w), \boldsymbol{\xi}_i \cdot \boldsymbol{n}_w > 0, \tag{42}$$



(a) (b)

FIG. 4. Mesh for flow around an oscillating circular cylinder: (a) full domain and (b) near the circular cylinder surface.

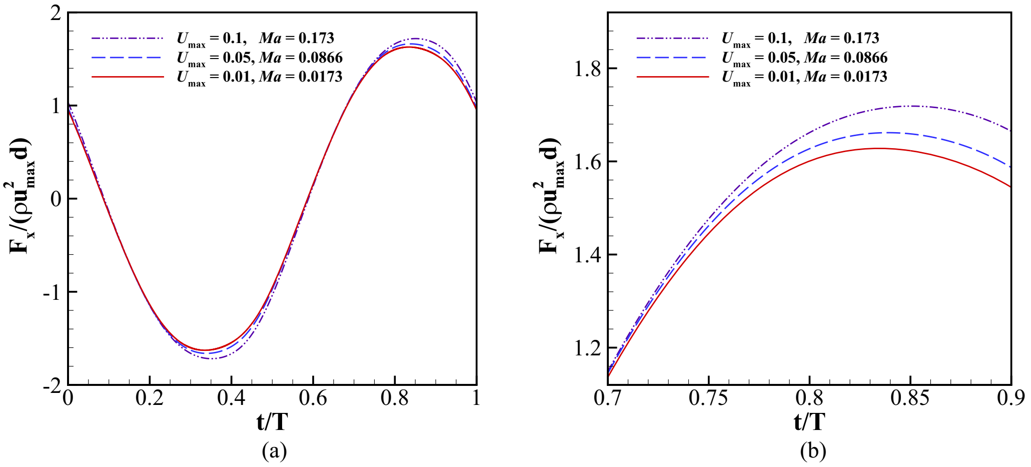


FIG. 5. Dimensionless inline force $F_x$ at different $U_{\max}$ ($\Delta t = 0.05$, $\Delta x = d/256$): (a) the full oscillating period $T$ and (b) part of the enlarged view.

where $\boldsymbol{n}_w$ represents a unit vector in the direction normal to the wall and with a point to the cell. Density at the wall $\rho_w$ is determined by the condition that no particles can go through the wall:

$$\sum_{\boldsymbol{\xi}_i\cdot\boldsymbol{n}_w>0}(\boldsymbol{\xi}_i-\boldsymbol{u}_w)\cdot\boldsymbol{n}_w f_w^{\mathrm{eq},n+1/2}(\boldsymbol{\xi}_i;\rho_w,\boldsymbol{u}_w) + \sum_{\boldsymbol{\xi}_i\cdot\boldsymbol{n}_w<0}(\boldsymbol{\xi}_i-\boldsymbol{u}_w)\cdot\boldsymbol{n}_w f_w^{n+1/2}=0. \tag{43}$$

Then it can be calculated as

$$\rho_w = -\frac{\sum_{\boldsymbol{\xi}_i\cdot\boldsymbol{n}_w<0}(\boldsymbol{\xi}_i-\boldsymbol{u}_w)\cdot\boldsymbol{n}_w f_w^{n+1/2}}{\sum_{\boldsymbol{\xi}_i\cdot\boldsymbol{n}_w>0}(\boldsymbol{\xi}_i-\boldsymbol{u}_w)\cdot\boldsymbol{n}_w f_w^{\mathrm{eq},n+1/2}(\boldsymbol{\xi}_i;1,\boldsymbol{u}_w)}, \tag{44}$$

where the distribution functions $f_w^{n+1/2}$ with direction $\boldsymbol{\xi}_i\cdot\boldsymbol{n}_w<0$ can be constructed following the procedure described in Sec. II B 1.

In this study, for the test cases of flow around obstacles, the far-field boundary condition will be used. Similar to the treatment of the wall boundary, the distribution function in

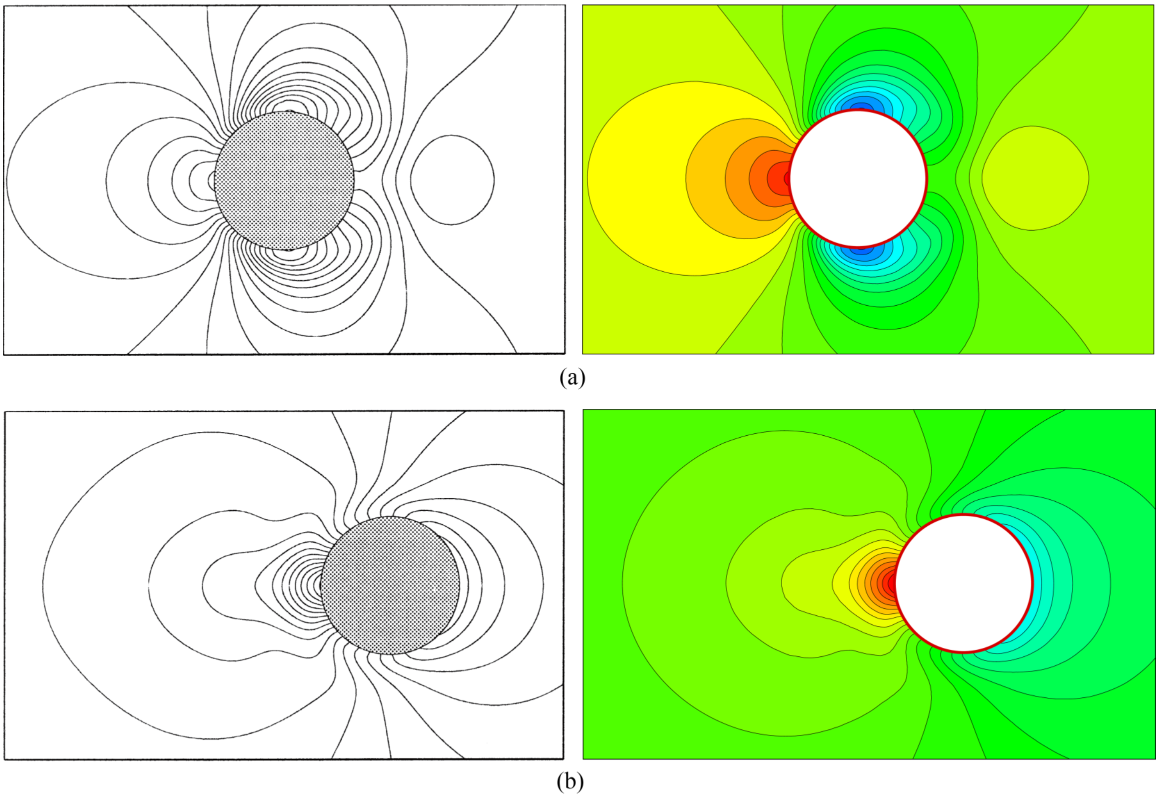
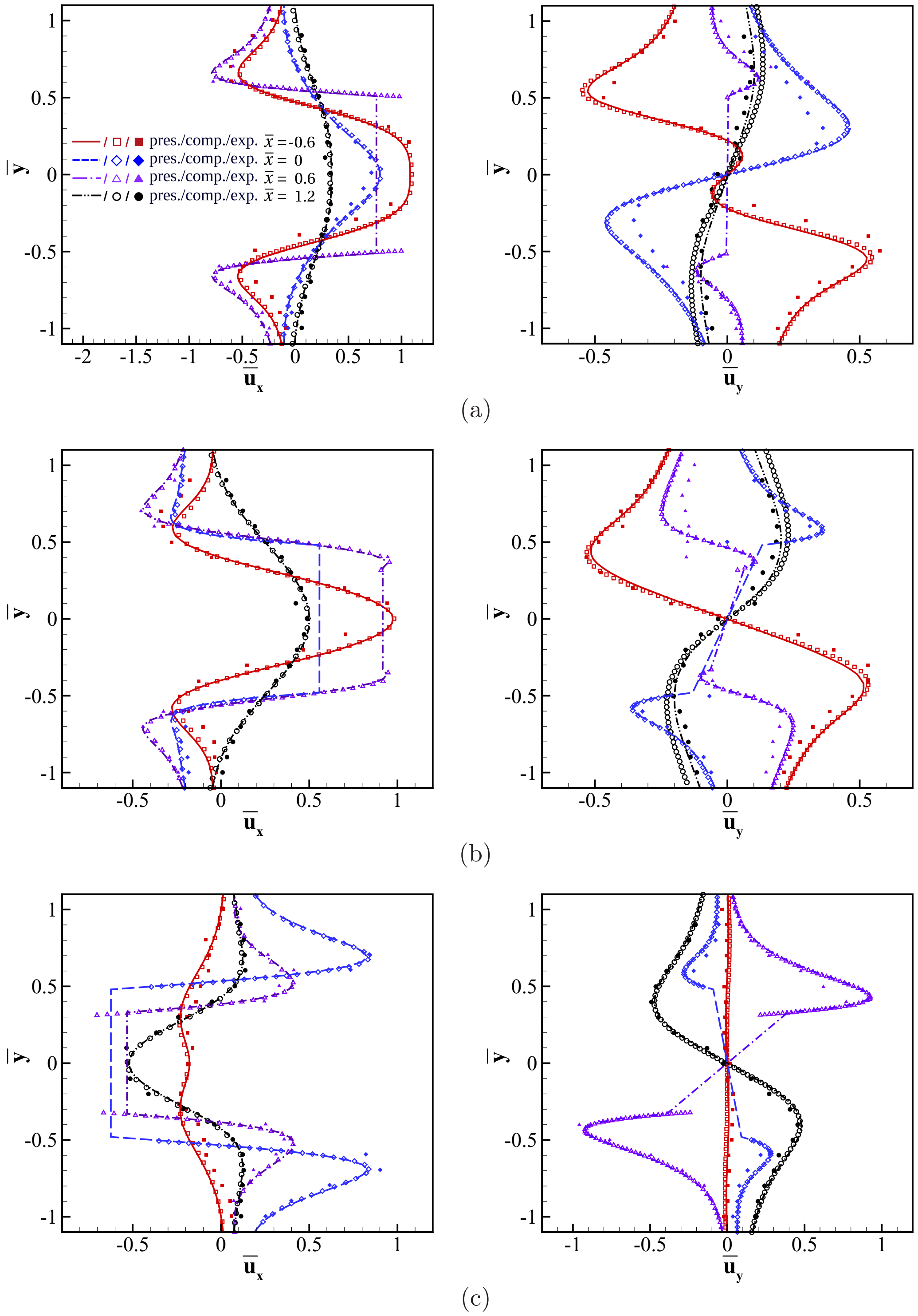


FIG. 6. Pressure isolines at phases (a) 0° and (b) 288°, Dütsch *et al.* [36] (left) and present (right) results.



(a)

(b)

(c)

FIG. 7. Velocity profiles at four vertical cross sections $\bar{x} = -0.6, 0, 0.6, 1.2$ for phase positions (a) 180°, (b) 210°, and (c) 330°. The present results are compared with the computational and experimental values of Dütsch *et al.* [36].

directions reflecting from the boundary can be calculated as

$$f^{n+1/2}(\boldsymbol{\xi}_i) = f^{\mathrm{eq}}(\boldsymbol{\xi}_i; \rho_0, \boldsymbol{u}_0),\ \boldsymbol{\xi}_i \cdot \boldsymbol{n}_w > 0, \tag{45}$$

where $\rho_0$ and $\boldsymbol{u}_0$ are the density and velocity of a free-stream, respectively.

## III. NUMERICAL RESULTS AND DISCUSSION

In this section, several test cases are considered to validate the proposed ALE-type DUGKS. The first case is a uniform flow for the GCL-compliant test, showing the importance of implementing a GCL-compliant scheme. The second and third cases test the application of the proposed method for

TABLE II. Drag coefficient $c_d$ and added mass coefficient $c_i$ for the flow around a circular cylinder oscillating in the stationary fluid.

| Work | $c_d$ | $c_i$ |
|---|---|---|
| Dütsch *et al.* [36][a] | 2.09 | 1.45 |
| Uzunoğlu *et al.* [39][a] | 2.10 | 1.45 |
| Yuan *et al.* [17] | 2.10 | 1.47 |
| Present | 2.10 | 1.48 |

[a]Data are obtained using the finest meshes in Refs. [36] and [39].

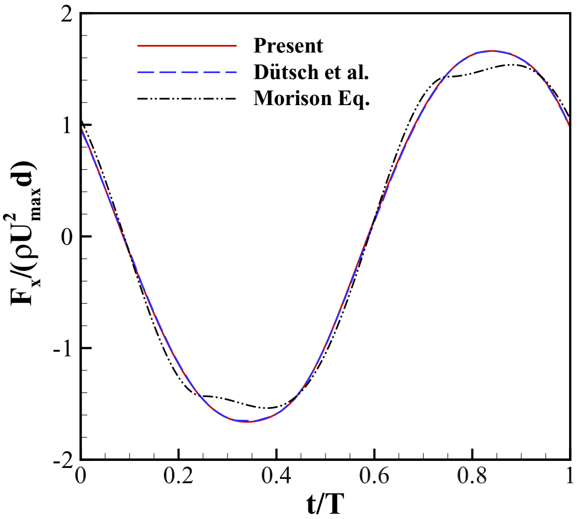


FIG. 8. Comparison of inline force $F_x$ with the result of Dütsch *et al.* [36] and empirical result obtained by Morison equation [38] [Eq. (50)].

low-speed continuum flows around an oscillating circular cylinder and around a pitching NACA0012 airfoil, respectively; they confirm that the proposed method can achieve better results compared with that of macromethods. The fourth and fifth cases test the application of the proposed method for two low-speed rarefied flow conditions: one is a moving piston driven by a rarefied gas and another is the flow caused by a plate oscillating in a direction normal to the boundary, which is a typical problem in microelectromechanical system (MEMS) devices but lacks systematic research [33]. Finally, a test case is also conducted for a rarefied flow around an oscillating circular cylinder, which is also a topic not researched extensively. For the continuum flow test cases, only the $g$ (density) distribution function is solved, and a three-point

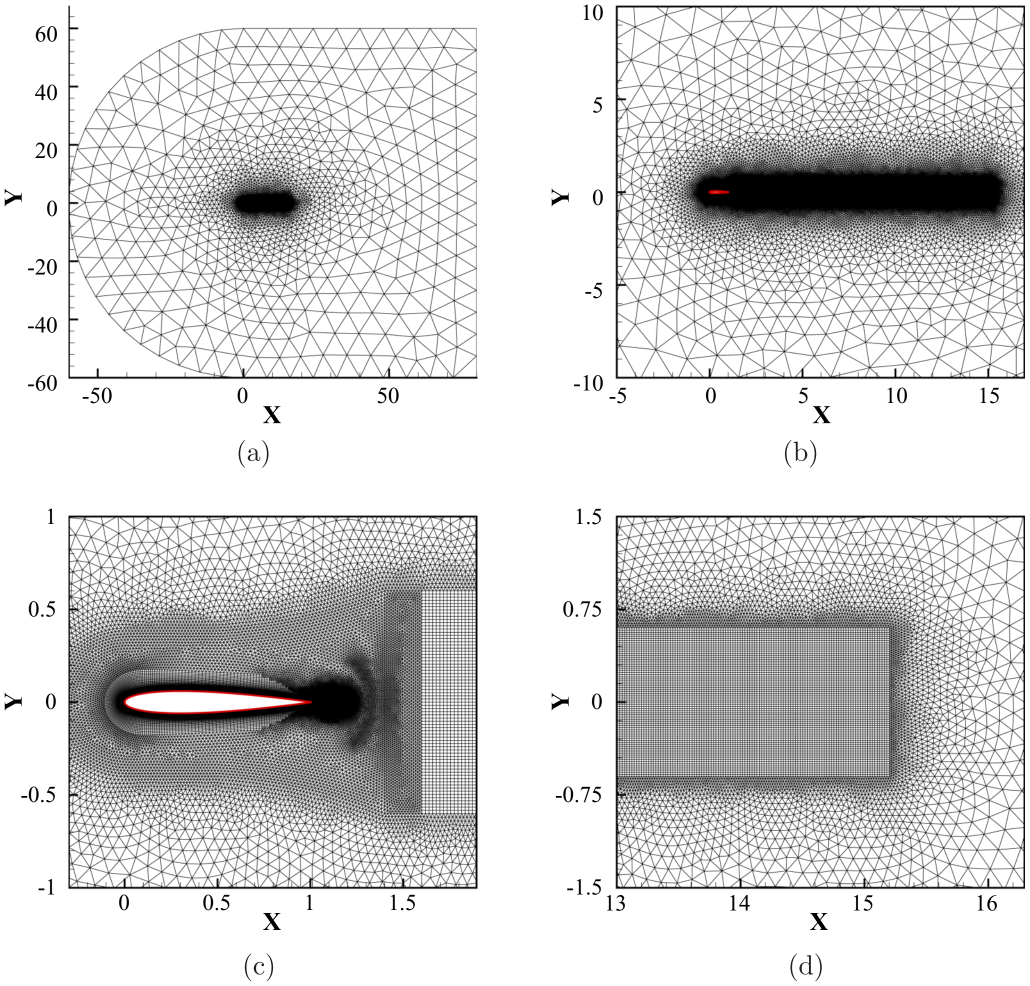


FIG. 9. Mesh for flow around a pitching NACA0012 airfoil: (a) full domain and (b), (c), (d) parts of the enlarged view near the airfoil.

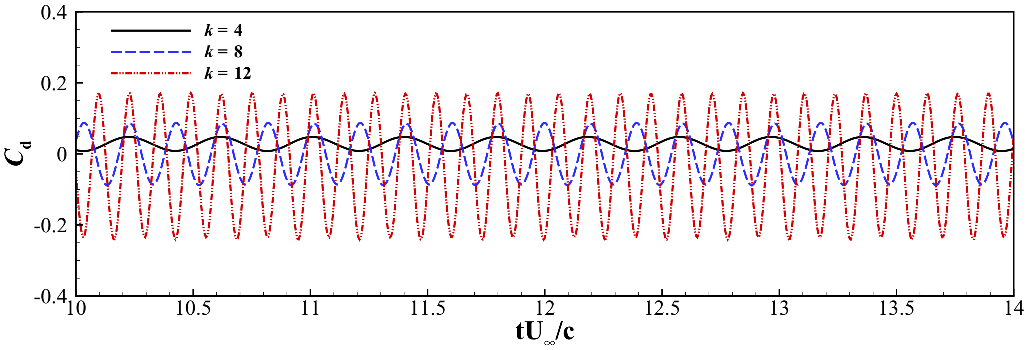


FIG. 10. Time evolutions of drag coefficient at different reduced frequencies $k$ for $d = 2°$.

Gauss–Hermite quadrature rule [8] (the D2Q9 lattice model in the LBM) is used to calculate the macroquantities. For the rarefied flow test cases, both $g$ (density) and $h$ (energy) distribution functions are solved, and the Gauss–Hermit and Newton–Cotes quadrature rules [13] are used.

The ALE-type DUGKS formulation has been coded with the help of Code_Saturne [34], an open-source computational fluid dynamics software of Electricite De France (EDF), France (http://www.code-saturne.org/cms/). We appreciate the development team of Code_Saturne for their great works.

### A. GCL compliance test with uniform flow condition

The uniform flow condition is usually used as a basic case for the GCL-compliant test, while several additional test cases can be found in Ref. [22]. For this problem, the computational domain is a square of side equal to 20, and the number of grid cells is 40 × 40. The time step is set to $\Delta t = 0.2$, the initial velocity is $u = U_0 = 0.1$, $v = 0$, and the initial density is $\rho_0 = 1.0$. In addition, the far-field boundary condition presented in Sec. II B 3 is used in this case.

First, the importance of GCL is illustrated. For the mesh motion method, the grid points oscillate randomly around their original locations with an amplitude $\pm 0.5\Delta x$ ($\Delta x$ is the size of a grid cell). Figure 2 shows the meshes, respectively, at the initial state and at instantaneous states when the grid points start moving. It has been demonstrated in Sec. II B 2 that during the update of the distribution function, if Eqs. (34) and (35) are used, the DGCL will be satisfied automatically. Otherwise, if Eq. (36) or (39) is used without the volume constraint, then the DGCL will be violated. Figure 3 shows the pressure and velocity contours after 10 iterative calculations when a DGCL-violated scheme is implemented. Clearly, the pressure and velocity are severely polluted, preventing continuation of the simulation. Consequently, if the grid points are moved in extremely irregular ways, a large GCL error occurs, which must be eliminated.

Second, the performance of DGCL schemes used in the ALE-type DUGKS is verified. In this part, the grid points also oscillate randomly around their original locations with an amplitude $\pm 0.5\Delta x$, and the spatial errors for pressure and velocity are calculated at four different cell sizes, that is, with the number of grid cells equal to 20 × 20, 40 × 40, 80 × 80, and 160 × 160. As shown in Table I, $L_2$ and $L_\infty$ errors of all schemes are maintained at significantly small values after 1000 iterations, thus indicating that all the DGCL-compliant schemes can eliminate the GCL error very well.

As the deformation rate of a moving grid cell in above numerical tests is large and discontinuous, a large GCL error will be introduced in the computational domain if a DGCL-violated scheme is implemented. From our further tests, when the amplitude of random oscillation is dropped to $\pm 0.1\Delta x$, the

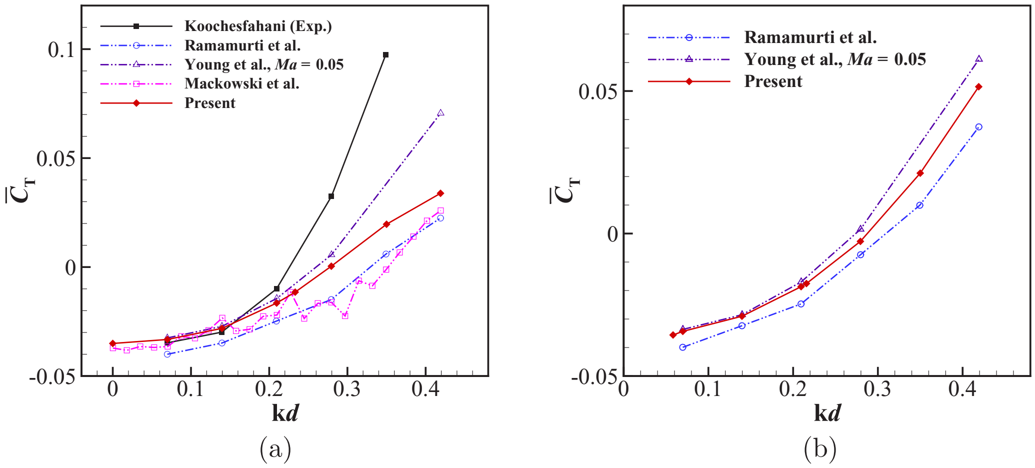
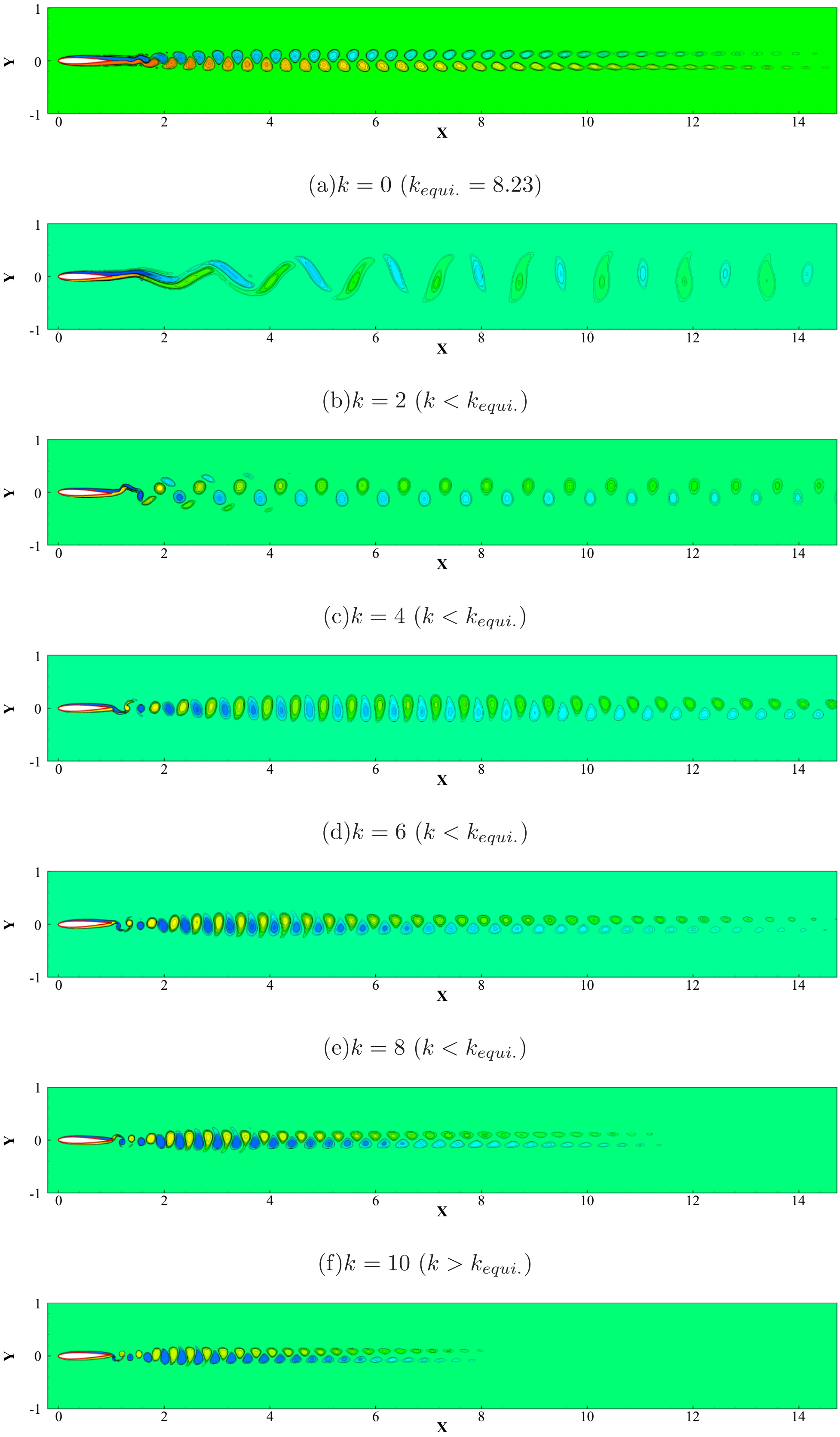


FIG. 11. Mean thrust coefficients $\bar{C}_T$ at different reduced frequencies $k$ for (a) $d = 2°$ and (b) $d = 4°$.



(a)$k = 0$ ($k_{equi.} = 8.23$)

(b)$k = 2$ ($k < k_{equi.}$)

(c)$k = 4$ ($k < k_{equi.}$)

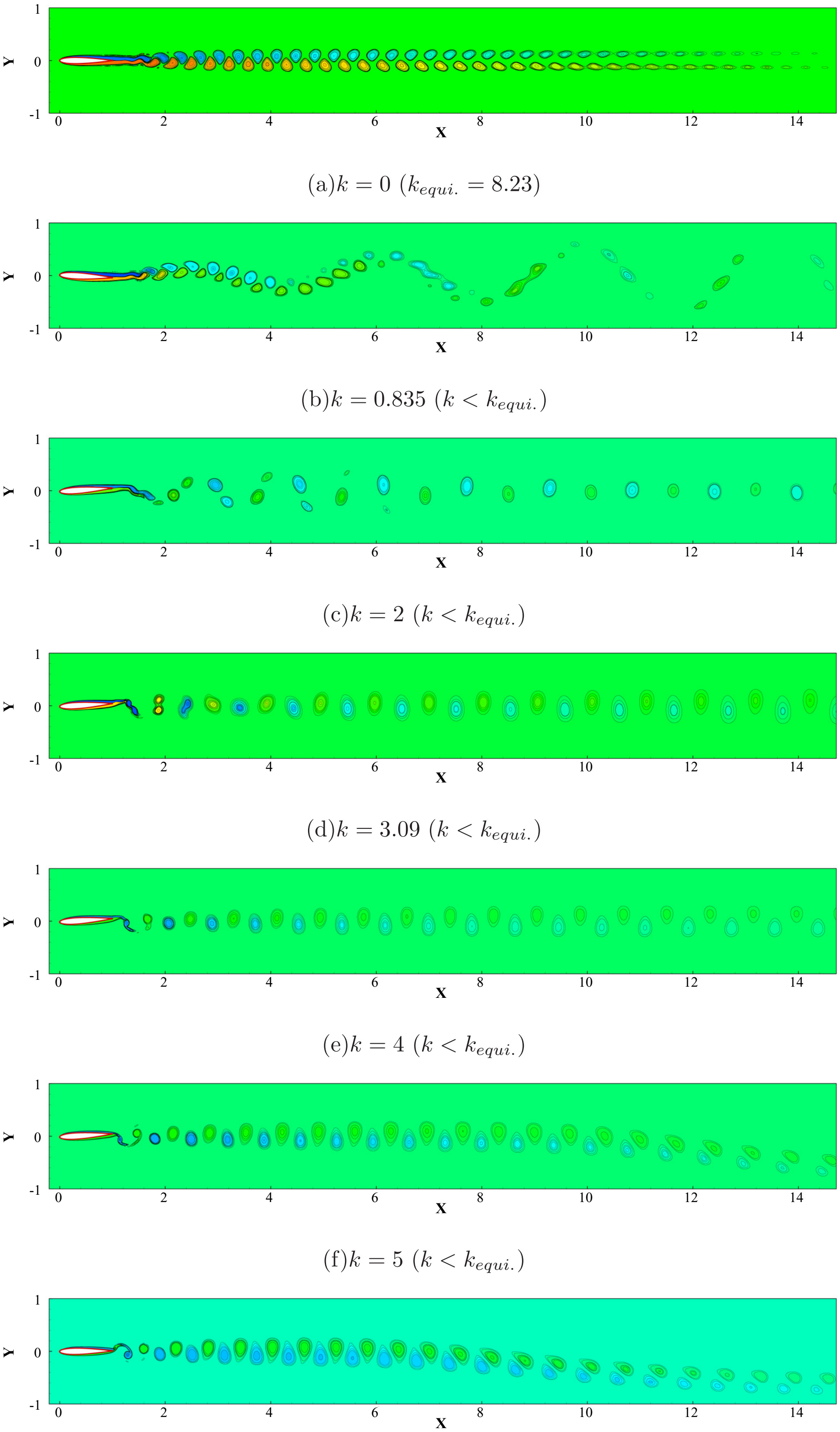
(d)$k = 6$ ($k < k_{equi.}$)

(e)$k = 8$ ($k < k_{equi.}$)

(f)$k = 10$ ($k > k_{equi.}$)

(g)$k = 12$ ($k > k_{equi.}$)

FIG. 12. Vortex shedding patterns at different reduced frequencies $k$ for $d = 2^{\circ}$.



(a)$k = 0$ ($k_{equi.} = 8.23$)

(b)$k = 0.835$ ($k < k_{equi.}$)

(c)$k = 2$ ($k < k_{equi.}$)

(d)$k = 3.09$ ($k < k_{equi.}$)

(e)$k = 4$ ($k < k_{equi.}$)

(f)$k = 5$ ($k < k_{equi.}$)

(g)$k = 6$ ($k < k_{equi.}$)

FIG. 13. Vortex shedding patterns at different reduced frequencies $k$ for $d = 4^\circ$.

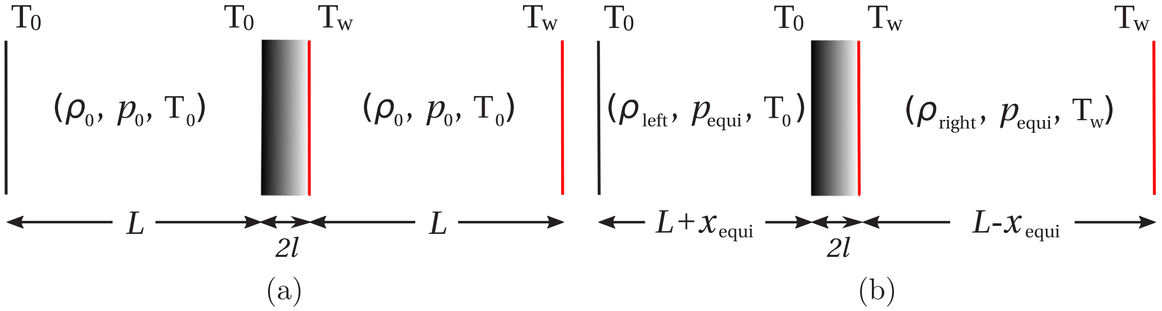


FIG. 14. Schematics of a moving piston driven by a rarefied gas: (a) initial stage and (b) equilibrium stage.

GCL error is considerably reduced. Consequently, by maintaining a regular grid point motion, a small GCL error can be achieved compared with other numerical errors. However, it has been reported that in some fields such as aero-elasticity analysis, the GCL error yields erroneous results [35]. Therefore, following the suggestions proposed in Ref. [22], the DGCL-compliant scheme is implemented in the following test cases.

### B. Continuum flow around an oscillating circular cylinder

In this section, the continuum flow around an oscillating circular cylinder is simulated. For this case, the cylinder oscillates sinusoidally in the horizontal direction, and the equation of motion can be expressed as

$$x(t) = -A\sin(2\pi ft), \tag{46}$$

where $x$ is the displacement of the cylinder around its equilibrium location, $A$ is the amplitude, and $f$ is the oscillating frequency. Following the set up described in Ref. [36], two key parameters that dominate the flow patterns, the Reynolds number Re and the Keulegan–Carpenter number $KC$, are defined. They are expressed as

$$\mathrm{Re} = \frac{\rho U_{\max} d}{\mu}, \tag{47}$$

$$\mathrm{KC} = \frac{U_{\max}}{fd}, \tag{48}$$

where $d$ is the diameter of the cylinder, $U_{\max}$ is the maximum motion velocity of the cylinder in the horizontal direction, $\rho$ is the fluid density, and $\mu$ is the fluid viscosity. In this simulation, we consider Re = 100 and KC = 5.

Figure 4 shows the mesh used in this case. The size of the computational domain is $[80d \times 60d]$, which is large enough to eliminate the influence of the far-field boundary condition. The total number of grid cells is approximately 49 000, with 240 points along the circular cylinder surface. Quadrangular cells are used to discretize the region near the cylinder surface with a width of $0.1d$, and triangle cells are used to discretize the other regions of the computational domain. At the initial stage, the cylinder is located at the center of the computational domain, with $\rho = 1.0$ and $u = v = 0$ to initialize the distribution functions. These values are also used as the far-field boundary conditions.

First, the numerical convergences of inline force $F_x$ at different Mach numbers Ma, time steps $\Delta t$, and minimum cell sizes $\Delta x$ are conducted. From our numerical tests, the influence of $\Delta x$ and $\Delta t$ on the results is found to be negligibly small; in the following simulations, $\Delta x = d/256$ and $\Delta t = 0.05$ will be used. Then, tests are conducted with Ma equal to 0.173 ($U_{\max} = 0.1$), 0.0866 ($U_{\max} = 0.05$), and 0.0173 ($U_{\max} = 0.01$) (considering the sound speed is equal to $1/\sqrt{3}$ for D2Q9 lattice model). As shown in Fig. 5, for an oscillating period $T$, the inline force $F_x$ is considerably

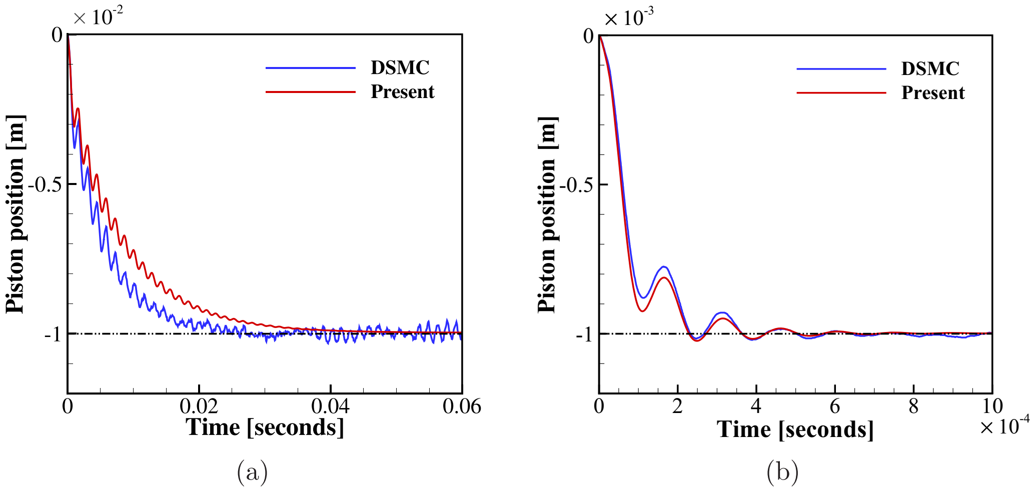


FIG. 15. Time histories of piston position at (a) Kn = 0.031 and (b) Kn = 0.31 (the black dash and dot lines represent the theoretical solutions).

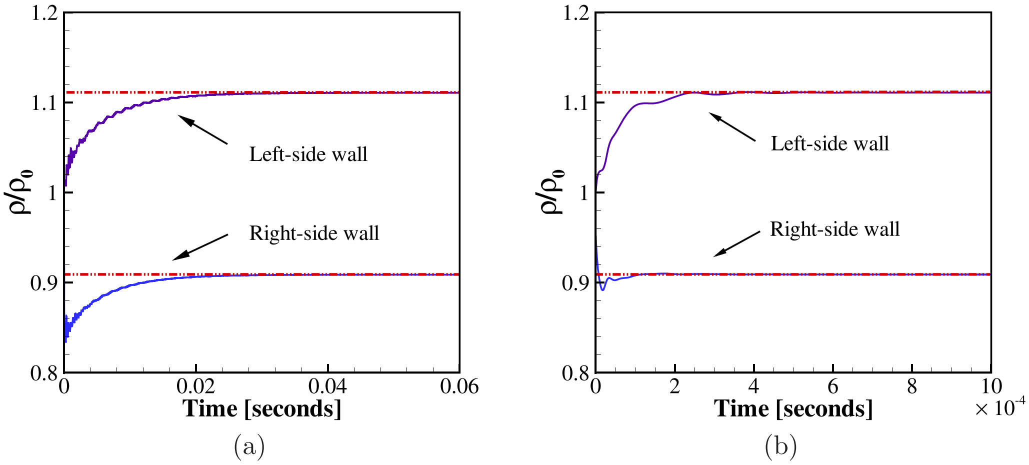


FIG. 16. Time histories of density fluctuation at the two walls of the piston for (a) Kn = 0.031 and (b) Kn = 0.31 (the red dash and dot lines represent the theoretical solutions).

influenced by Ma, especially for the amplitude of $F_x$. In the framework of the LBM based on the D2Q9 lattice model, the compressible Navier–Stokes equations are recovered through the Chapman–Enskog perturbation expansion. Therefore, to simulate an incompressible fluid flow with a compressible scheme, a numerical error related to the artificial compressibility will arise and may leads to some undesirable errors in the numerical simulations [37]. By considering the time costs of computation, Ma = 0.0866 will be used in the following simulations.

Second, other results are compared to further validate the proposed numerical method. Pressure isolines at phase positions 0° and 288° are shown in Fig. 6. These isolines agree well with the numerical results of Dütsch *et al.* [36]. Furthermore, the velocity profiles $\bar{u}_x$ and $\bar{u}_y$ at four vertical cross sections $\bar{x} = -0.6, 0, 0.6, 1.2$ for three phase positions 180°, 210°, and 330° are compared with the numerical and experiment results obtained by Dütsch *et al.* [36]; $\bar{u}_x$, $\bar{u}_y$, $\bar{x}$, and $\bar{y}$ are defined as

$$\bar{x} = \frac{x}{d}, \quad \bar{y} = \frac{y}{d}, \quad \bar{u}_x = \frac{u_x}{u_{\max}}, \quad \bar{u}_y = \frac{u_y}{u_{\max}}, \tag{49}$$

where $x$ and $y$ are the coordinates relative to the equilibrium position of the oscillating cylinder, $u_x$ and $u_y$ are the velocity components in the horizontal and vertical directions. As shown in Fig. 7, our results also generally agree well with the numerical and experiment results of Dütsch *et al.* [36].

For an oscillating flow, the semiempirical equation of Morison *et al.* [38] is widely used to estimate $F_x$ on a body. When a circular cylinder oscillates in the stationary fluid, the time-dependent $F_x$ can be expressed as

$$F_x = -\tfrac{1}{2}\rho d c_d \dot{x}|\dot{x}| - \tfrac{1}{4}\pi \rho d^2 c_i |\ddot{x}|, \tag{50}$$

where $x$ is the displacement of the cylinder, and $c_d$ and $c_i$ are the drag coefficient and the added mass coefficient, respectively. By integrating the pressure and stress along the surface of the circular cylinder, $F_x$ is calculated. Then, using least-squares fitting or Fourier analysis, $c_d$ and $c_i$ are evaluated. The fitted $c_d$ and $c_i$ are compared with other author's numerical results shown in Table II. Although $c_i$ is slightly higher, the proposed ALE-type DUGKS generally achieves good results. Given the values of $c_d$ and $c_i$, Eq. (50) can be used to evaluate the empirical values of $F_x$ in an oscillating period. As shown

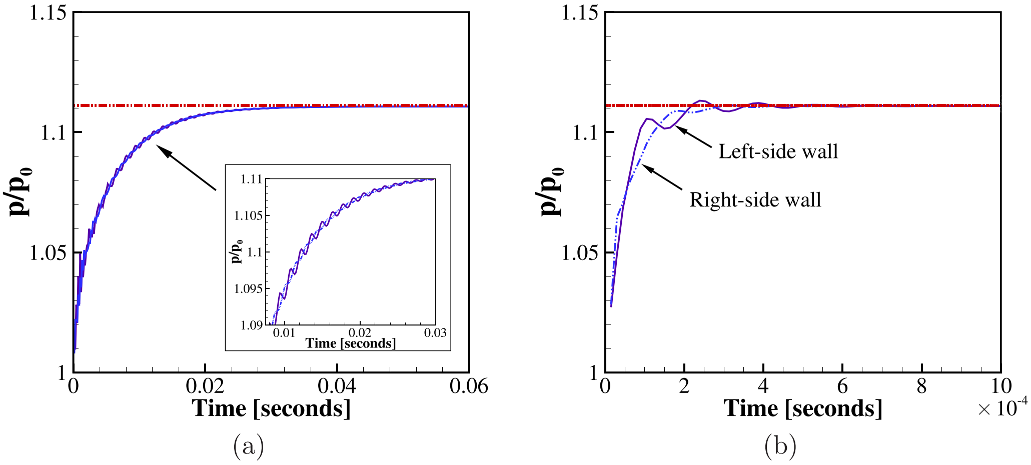


FIG. 17. Time histories of pressure fluctuation at the two walls of the piston for (a) Kn = 0.031 and (b) Kn = 0.31 (the red dash and dot lines represent the theoretical solutions).

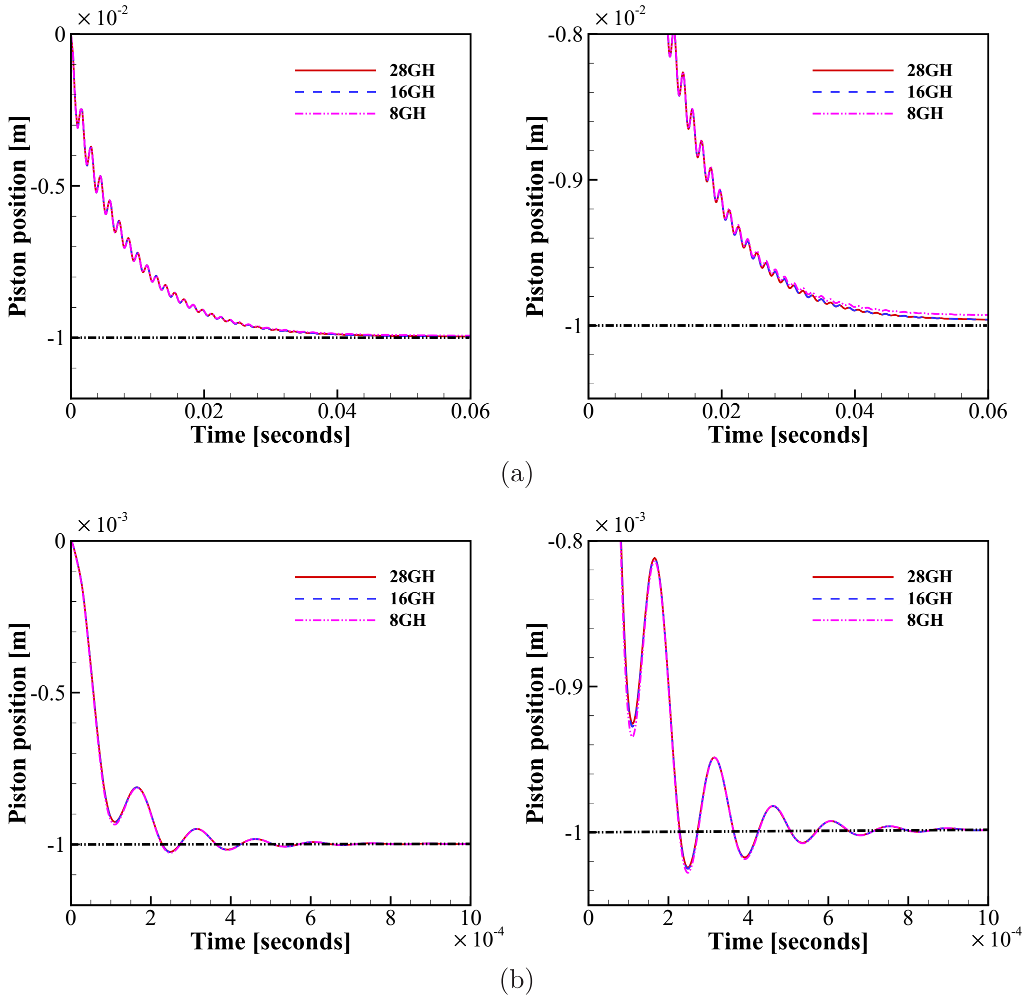


FIG. 18. Convergence tests based on different number of abscissas at (a) Kn = 0.031 and (b) Kn = 0.31 (the black dash and dot line represent the theoretical solution). 28GH: 28-abscissas Gauss–Hermit quadrature rule.

in Fig. 8, our result agrees well with that of Dütsch *et al.* [36] and is approximately consistent with the empirical values of Morison *et al.* [38].

### C. Continuum flow around a pitching NACA0012 airfoil

To study the details of propulsion in case of insects, birds, fishes, etc., the flow around an oscillating airfoil with a pitching or heaving motion or both is usually used as the benchmark test case [40]. In this section, the pitching motion of a NACA0012 airfoil around its quarter-chord point is considered to verify the performance of the proposed ALE-type DUGKS in simulating the rotary motion boundary problems. The variation in angle of attack (AOA) $\alpha$ for a pitching airfoil can be expressed as

$$\alpha = d\sin(2\pi ft), \qquad (51)$$

where $d$ is the amplitude of the pitching motion and $f$ is the pitching frequency. Usually, a new parameter, reduced frequency $k$, is defined for the pitching airfoil. The relationship between $k$ and $f$ is $k = 2\pi fc/U_0$, where $c$ is the airfoil chord length and $U_0$ is the velocity of free-stream. The Reynolds number Re based on $c$ is set at 12 000. The flow at this Reynolds number can be generally treated as a laminar flow [41]. Figure 9 shows the mesh used in this case. The total number of grid cells is 140 831, with 291 points at the airfoil surface. The region near the airfoil is discretized into quadrangular cells, with the minimum size of grid cells being $1.0 \times 10^{-4}c$. In addition, the mesh in the region with approximately $15c$ length behind the airfoil trail is refined with the Cartesian grid cells to capture the vortex streets. The number of grid cells used in this case is slightly large, so if only some results such as the force coefficient are considered, approximately $50 \sim 60 \times 10^3$ grid cells sufficiently provide high-accuracy results. In addition, as shown in Sec. III B, a small value of $U_0$ ($U_0 = 0.05$) must be used to eliminate the compressible effect.

First, the flow around a stationary airfoil is simulated. As experimentally revealed by Koochesfahani [42], the phenomenon of vortex shedding can be observed at the Reynolds number of 12 000, and the equivalent reduced frequency $k_{\text{equi.}}$ based on the vortex shedding frequency is 8.7. Our numerical result shows $k_{\text{equi.}} = 8.23$, which is close to the experiment value. In addition, the mesh in the region with approximately $0.5c$ length behind the airfoil trail must be refined. If the mesh in this region is too coarse, such as that of

O-type mesh, a steady flow will be obtained by the numerical simulation.

Second, a series of flows around a pitching airfoil is simulated at $d = 2^\circ$ and $4^\circ$, respectively, and at different reduced frequencies $k$. Figure 10 shows the time evolutions of the drag coefficient $C_d$ at three values of $k$. At a small value of $k$ ($k = 4$), both the maximum and the minimum $C_d$ values are greater than zero, which generates a drag force in the flow. When the $k$ value is increased ($k = 8$), the minimum of $C_d$ is less than zero. Although the flow still generates the drag force, the magnitude of mean $C_d$ reduces. When $k$ is continually increased ($k = 12$), the absolute value of minimum $C_d$ becomes greater than the maximum $C_d$, which generates a thrust force in the flow. Here, a new force coefficient, $C_T$, is defined, which represents the thrust coefficient and is given as $C_T = -C_d$. Figure 11 shows the mean thrust coefficient $\bar{C}_T$ at different $kd$. In general, the same tendencies are observed as those of other numerical results [40,43,44] and experimental results [42]. For $d = 2^\circ$, when $kd$ is less than 0.2, the numerical results are consistent with each other as well as with the experiment values. However, when $kd$ is greater than 0.2, the discrepancy increases, which can be explained by some reasons presented in Ref. [43].

Figures 12 and 13 show the vortex shedding patterns at $d = 2^\circ$ and $d = 4^\circ$, respectively. For comparison, the vortex shedding pattern at $k = 0$ (stationary airfoil) is also presented. For $d = 2^\circ$, at a small value of $k$, the sizes of vortices are considerably larger than those generated by a stationary airfoil. When $k$ increases, as the thrust force is generated, the original two rows of the vortex street appears to merge into one row. Moreover, when $k < k_{\text{equi.}}$, even though the region with approximately $15c$ length is away from the airfoil trail, clear structures of the vortex street are maintained well. Otherwise, these structures quickly dissipate at large values of $k$. For $d = 4^\circ$, a form of undulating vortex sheet is generated at $k = 0.835$, whereas at $k = 3.09$, a double-vortex pattern is observed, but this structure seems to dissipate and merge into one another at approximately $2c$ length behind the airfoil trail. For $k = 2$, which is between 0.835 and 3.09, the length of the undulating vortex sheet is reduced (about $5c$ behind the airfoil trail), and the double-vortex features are generated but not very clear. At large values of $k$, double-vortex features disappear, and the vortex streets that are almost a straight line at small values of $k$ are deflected. Similar patterns of vortex street also illustrated by Liang *et al.* [41] with a numerical simulation and by Koochesfahani [42] with an experiment.

### D. A moving piston driven by a rarefied gas

In this section, the case of a moving piston driven by a rarefied gas is simulated. This problem has been studied by Dechristé *et al.* [45] with a deterministic numerical scheme coupled with the immersed boundary method, and by Shrestha *et al.* [6] with the DSMC. Figure 14 shows the schematics of this problem. The one-dimensional computational domain is divided into two subdomains by a piston, with the length of each subdomain being $L$ and the width of the piston being $2l$. At the initial stage, the two subdomains are filled with the same gas, with its density $\rho$, pressure $p$, and temperature $T$ set at the same values. For the right part of the computational domain, as the temperature of wall is set higher than that of the gas, the gas pressure increases, which pushes the piston from right to left. Finally, the piston stops moving when the pressures at the two walls of the piston are the same. By considering the law of conservation of mass and the state equation of gas in each part, we have the following [45]:

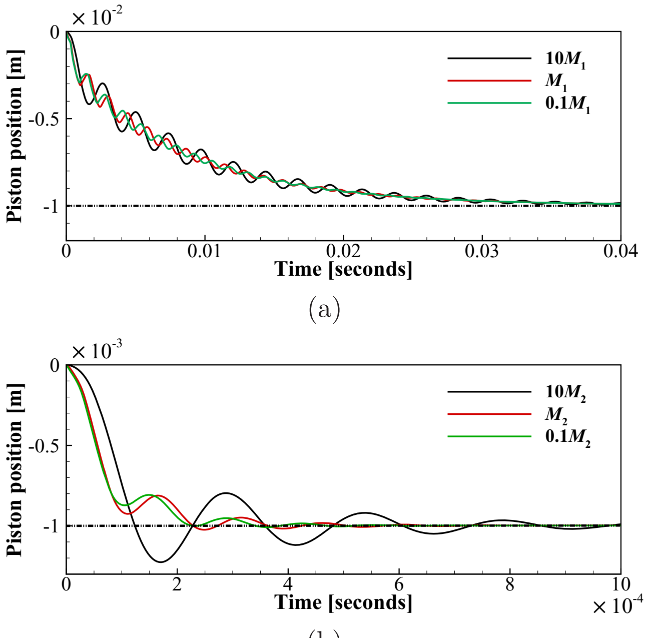
$$\rho_0 L = \rho_{\text{left}}(L + x_{\text{equi.}}),\ \ \rho_0 L = \rho_{\text{right}}(L - x_{\text{equi.}}), \tag{52}$$

and

$$\rho_{\text{left}} R T_0 = p_{\text{equi.}},\ \ \rho_{\text{right}} R T_w = p_{\text{equi.}}, \tag{53}$$

where $R$ is the gas constant. Then, the equilibrium location of the piston can be calculated as

$$x_{\text{equi.}} = L\frac{1 - T_w/T_0}{1 + T_w/T_0}; \tag{54}$$



(a)

(b)

FIG. 19. Comparisons of time history of piston position at different masses for (a) Kn = 0.031 and (b) Kn = 0.31 (the black dash and dot line represent the theoretical solution).

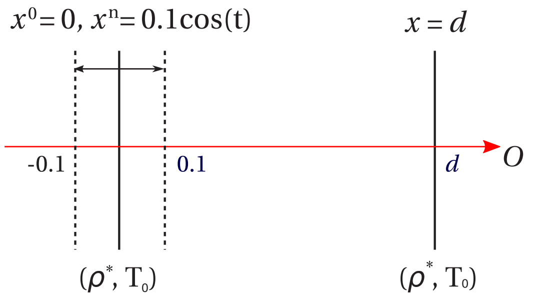
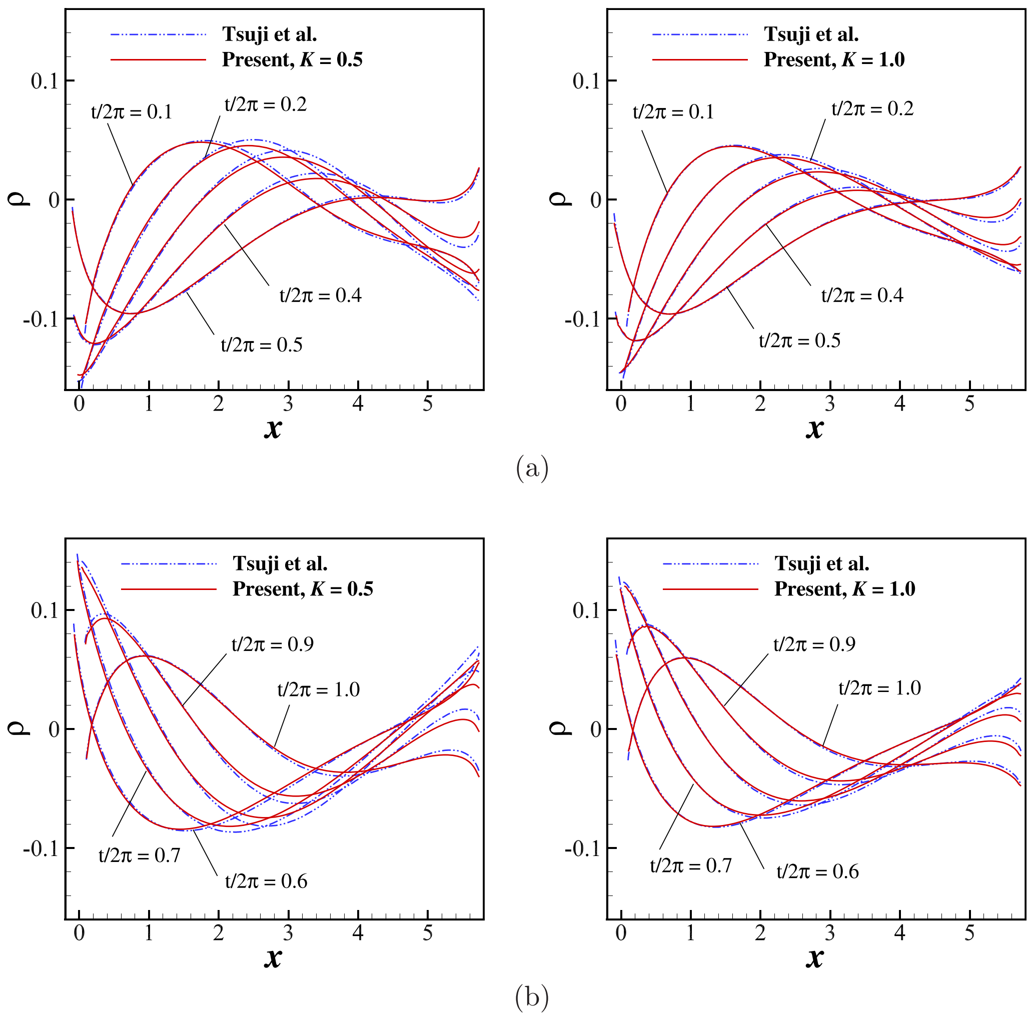


FIG. 20. Schematic of the rarefied flow caused by a moving plate oscillating in its normal direction.



(a)

(b)

FIG. 21. Profiles of density at (a) $t/2\pi = 0.1, \dots, 0.5$ and (b) $t/2\pi = 0.6, \dots, 1.0$ and at two $K$ numbers.

the pressure and density at the equilibrium state for each part are

$$p_{\text{equi.}} = \frac{L}{L + x_{\text{equi.}}} p_0, \quad \rho_{\text{left}} = \frac{p_{\text{equi.}}}{RT_0}, \quad \rho_{\text{right}} = \frac{p_{\text{equi.}}}{RT_w}. \tag{55}$$

Following the set up described by Shrestha *et al.* [6], the gas is argon, with atomic mass $m_g = 6.63 \times 10^{-26}$ kg, and atom diameter $3.68 \times 10^{-10}$ m under a hard-sphere collision model. The initial values for $T_0$, $T_w$, and $p_0$ are set to 270 K, 330 K, and 10 Pa, respectively. The initial velocities for each part are set to zero. By the ideal gas law, the initial density can also be calculated. In this test case, two computational geometries, $L = 0.1$ m, $l = 0.01$ m, and $L = 0.01$ m, $l = 0.001$ m, are considered. Consequently, based on the piston width ($2l$) and the given condition of the gas, the flows at two Knudsen (Kn) numbers, Kn = 0.031 and Kn = 0.31, are simulated. The mass of the piston $M$ is set to $M_1 = 3.56 \times 10^{-6}$ kg and $M_2 = 3.56 \times 10^{-7}$ kg at the two computational conditions, which are equal to 1/10 of the total mass of the gas in the computational domain. In addition, 400 grid cells are used to discretize the computational domain. Finally, the velocity $u_p$ and location $x_p$ of the piston at the $n + 1$ time step can be calculated as

$$u_p^{n+1} = u_p^n + \Delta t \times \frac{S}{M}\left(p_l^n - p_r^n\right), \quad x_p^{n+1} = x_p^n + \Delta t \times u_p^n, \tag{56}$$

where $\Delta t$ is the time step used in the fluid numerical simulation, $S$ is the area of the piston, and $p_l$ and $p_r$ are the pressure acting on the two walls of the piston.

Figure 15 shows the time evolutions of the piston location compared with the DSMC results [6] at two Kn numbers. It can be observed that the position of the piston at Kn = 0.31 converges much faster with its equilibrium position than that at Kn = 0.031. Although the convergence processes are different, the time required for the piston to reach its equilibrium position is almost the same, which is approximately 0.03 s at Kn = 0.031 (about $0.975x_{\text{equi.}}$ for present result at that time) and 0.0005 s at Kn = 0.31. In addition, for the result of the DSMC at a small value of Kn, the piston fluctuates around that equilibrium position, and several independent runs are needed to reduce this stochastic fluctuations. On the contrary, our results are very smooth even at a small value of Kn. Moreover, the black dash and dot lines shown in figures are the theoretical solutions [Eq. (54)], and the errors between the numerical and theoretical solutions are less than 1% for

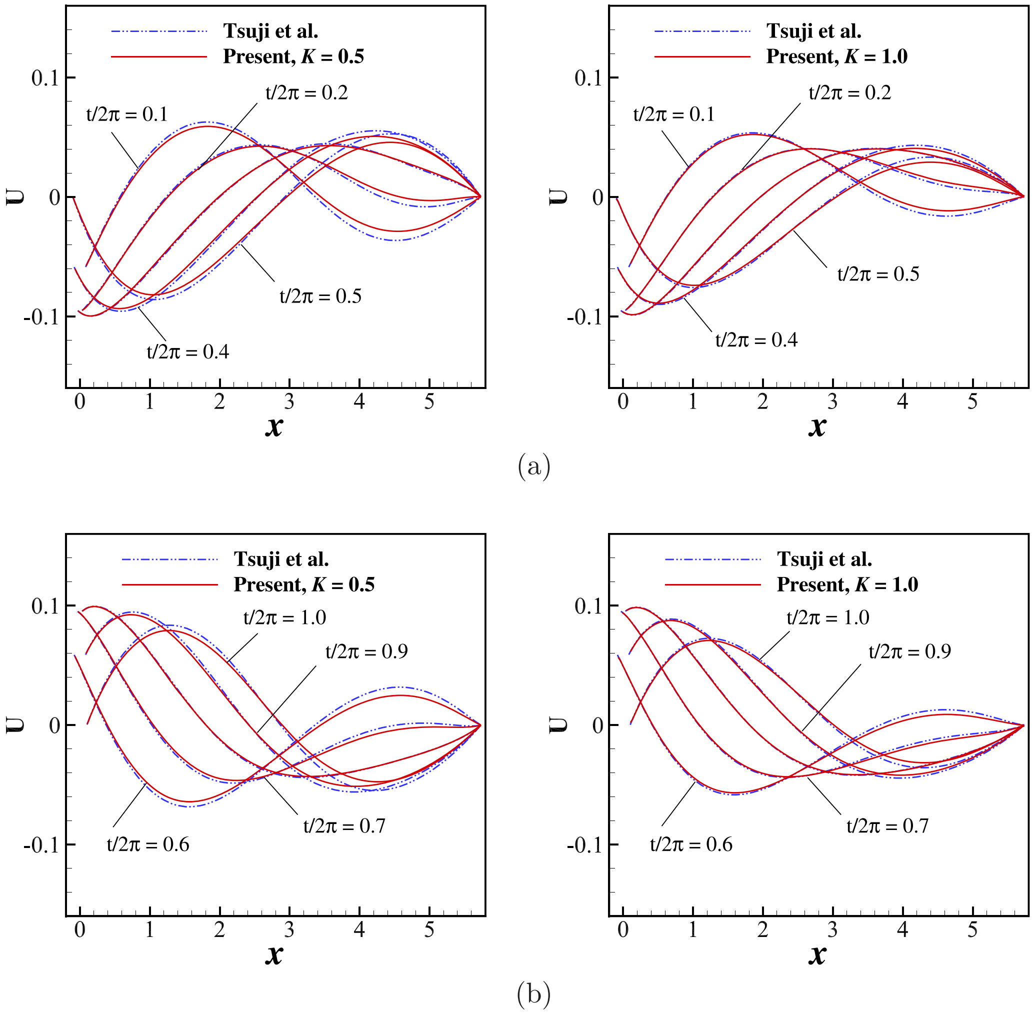


FIG. 22. Profiles of velocity at (a) $t/2\pi = 0.1, \ldots, 0.5$ and (b) $t/2\pi = 0.6, \ldots, 1.0$ and at two $K$ numbers.

both values of Kn. Figures 16 and 17 show the time evolutions of density and pressure at the two walls of the piston, respectively. Our numerical results are also consistent with the theoretical solutions. During the evolutions, the pressure difference of the two walls of the piston at $\mathrm{Kn} = 0.31$ is larger than that at $\mathrm{Kn} = 0.031$, which may also indicate that the piston at a large value of Kn moves faster to its equilibrium position.

In this case, the Gauss–Hermit quadrature rule is used [13] to integrate the macroquantities, and the code to calculate the abscissas and weights is presented in Ref. [46]. Figure 18 shows the convergence tests based on the different number of abscissas. From our test, for $\mathrm{Kn} = 0.031$, the integral accuracy based on 28 abscissas is good enough to achieve a convergent result. However, for $\mathrm{Kn} = 0.31$, 8 abscissas may also sufficiently provide a good result. Unexpectedly, in this test case, less discrete velocities are needed at a large value of Kn, which might entail further research to explain this observation. In addition, the motion of the piston at different masses is also considered. As Fig. 19 shows, the effect of the fluid structure interaction at a large mass is obvious, especially at a large value of Kn. Further, at a small value of Kn, although the amplitudes of each oscillation period of the piston are different, the average positions are almost the same during the convergence process. Besides, the oscillation of the piston at a small mass will decay faster than that at a large mass.

### E. Rarefied flow caused by a plate oscillating in its normal direction

This section presents the simulation of another rarefied flow test case presented in Ref. [33]. Figure 20 shows the schematic of this problem. In a one-dimensional domain, the right wall remains stationary and the left wall oscillates with the cosine function $x(t) = a_w \cos(\omega t)$ ($a_w = 0.1$ and $\omega = 1$ in this case). The amplitude of the moving wall is set into a small value to ensure that this case is a low-speed flow. Following the set up described in Ref. [33], in the initial stage, the length of the computational domain is $d = 2\pi\sqrt{5/6}$, which is the wavelength of the sinusoidal acoustic wave with an angular frequency $\omega$ in an inviscid (Euler) gas [47]. The Knudsen number Kn is given as

$$\mathrm{Kn} = \frac{2}{\sqrt{\pi}} K, \tag{57}$$

where $K$ is the special Kn number, and two $K$ numbers, equal to 0.5 and 1.0, are considered in this case. $\rho_0 = 1.0$, $T_0 = 1.0$, and $u_0 = 0.0$ are used to initialize the distribution functions $g$

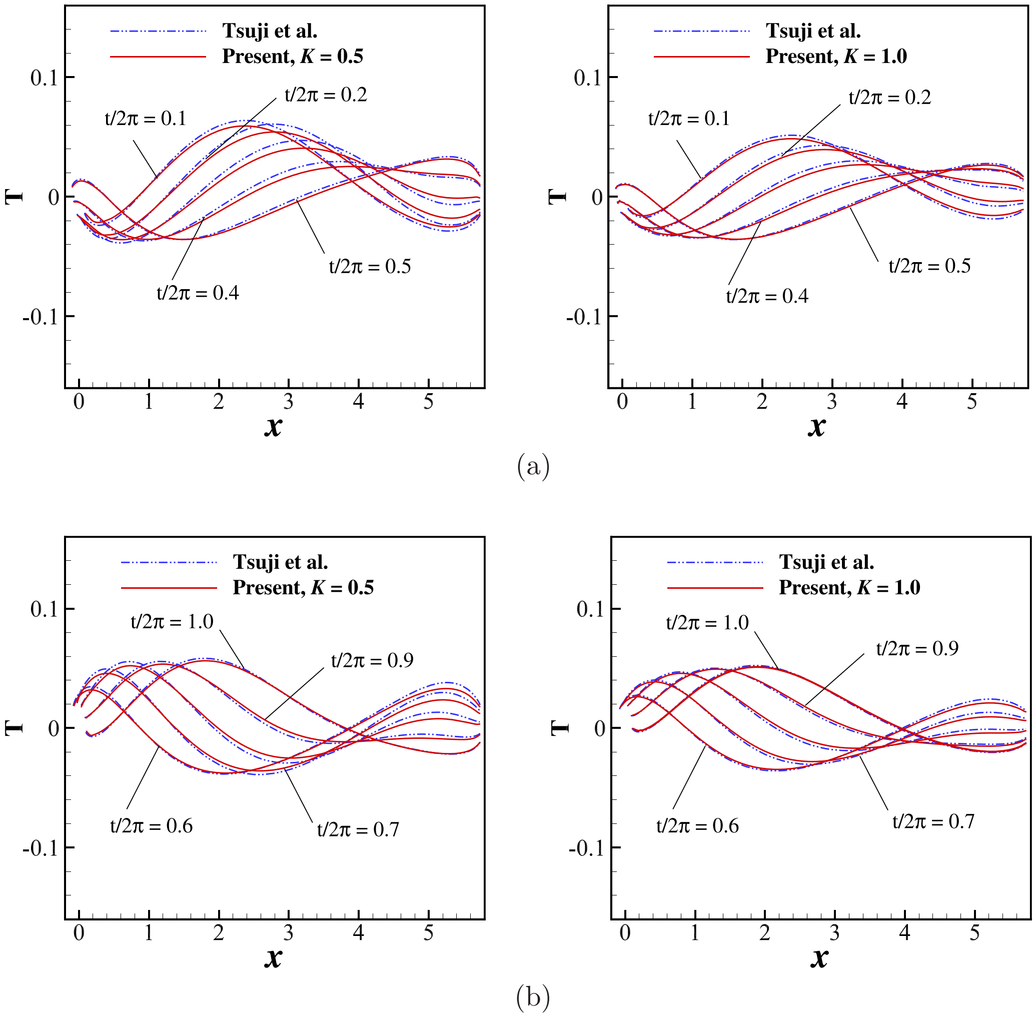


FIG. 23. Profiles of temperature at (a) $t/2\pi = 0.1, \ldots, 0.5$ and (b) $t/2\pi = 0.6, \ldots, 1.0$ and at two $K$ numbers.

and $h$. We use 400 grid cells to discretize the computational domain at $K = 0.5$ and 200 at $K = 1.0$. In addition, in each oscillating period ($T = 2\pi$), 32 000 iteration computations are implemented at $K = 0.5$ and 64 000 at $K = 1.0$.

Figures 21–23 show the profiles of density, velocity, and temperature, respectively, at ten moments in each oscillating period. Generally, our results agree well with that of Tsuji *et al.* [33], especially at a large value of $K$. However, without more references, it is difficult to identify which result is better. From these figures, it can be observed that at $t/\pi = 1.0$, the velocity profile is almost sinusoidal shape, but the profiles of density and temperature deviate from it significantly. Furthermore, the velocity profile deviates more from being sinusoidal shape and tends to attenuate rapidly at a large $K$ value, especially for the right part of the wave.

To integrate macroquantities in the quadrature rule, the Newton–Cotes rule is used. At a large $K$ value, since the number of particles is considerably decreased, the wave generated

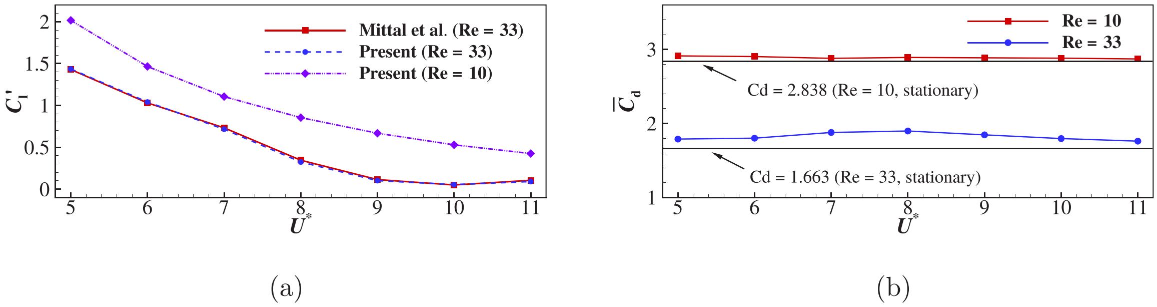
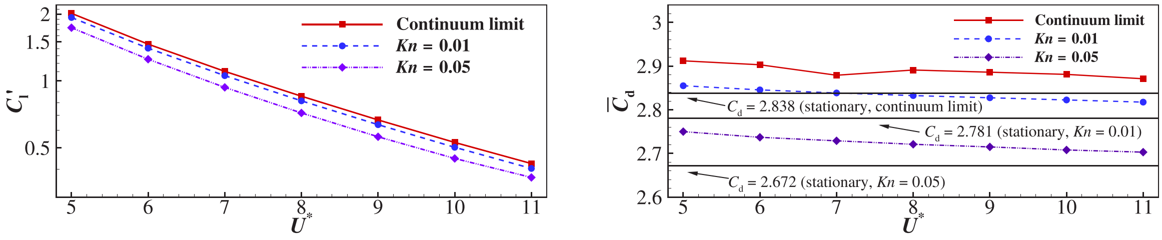


FIG. 24. The (a) maximum lift coefficient $C_l'$ and (b) mean drag coefficient $\bar{C}_d$ at two Reynolds numbers in a continuum flow regime.



(a) (b)

FIG. 25. The (a) maximum lift coefficient $C_l'$ and (b) mean drag coefficient $\bar{C}_d$ at two Kn numbers and at Re = 10.

from the left moving wall and that reflected from the right stationary wall lead to the singular distribution function [4]. Therefore, to obtain smooth results, the number of abscissas is increased considerably at a large $K$ value. From our tests, for $K = 0.5$, 50 abscissas are enough to obtain convergent and smooth results, whereas for $K = 1.0$, about 200 abscissas are needed to obtain smooth results.

### F. Uniform rarefied flow around an oscillating circular cylinder

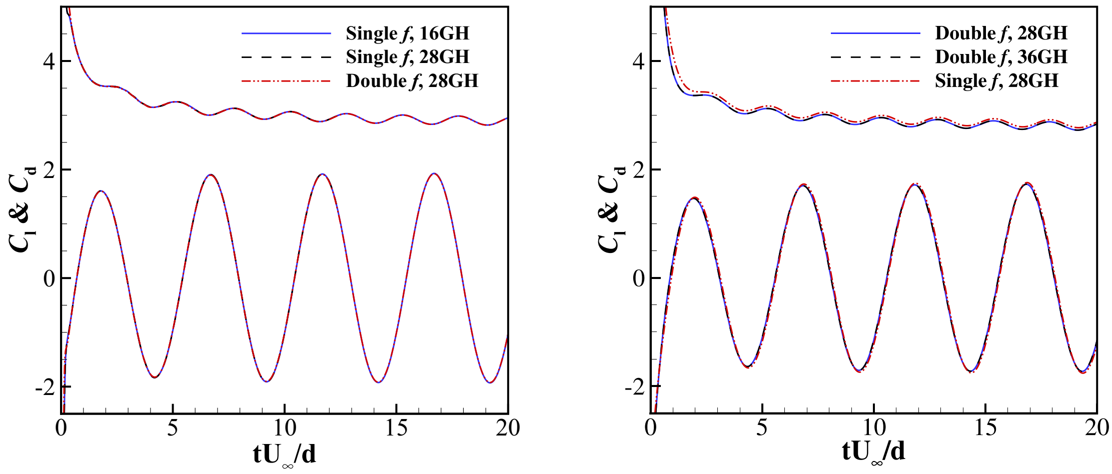
This section presents the simulation of an extended test case with a uniform flow around an oscillating circular cylinder; this condition is studied in detail in a continuum regime but insufficiently in a rarefied regime.

First, following the set up described by Mittal *et al.* [48], the continuum flow around an oscillating cylinder is simulated at a Reynolds number Re = 33. Different from the test case described in Sec. III B, in which the fluid is stationary at the initial stage, here a uniform flow is used as the initial condition. In addition, a similar hybrid mesh shown in Fig. 4 is also used in this case. The motion of a cylinder in the vertical direction can be expressed as

$$y(t) = A\sin\left(2\pi\frac{1}{d}\frac{U_\infty}{U^*}t\right), \tag{58}$$

where $y$ is the displacement of the cylinder around its equilibrium location, $d$ is the diameter of the cylinder, $A$ is the amplitude ($A = 0.25d$), $U_\infty$ is the velocity of free-stream, and $U^*$ is the reduced velocity of the cylinder ($1/U^*$ is the dimensionless oscillating frequency of cylinder). Then, $U^* = 5$–11 is considered in the simulation. As shown in Fig. 24(a), the forced oscillating cylinder generates a lift force, and the maximum lift coefficients $C_l'$ at different values of $U^*$ agree well with the results of Mittal *et al.* [48]. Moreover, the mean drag force $\bar{C}_d$ on the oscillating cylinder is slightly larger than that on a stationary cylinder [see Fig. 24(b)].

Second, the rarefied flow around an oscillating cylinder is simulated. To maintain the flow at a low speed, the Reynolds number is set to Re = 10, and the Knudsen number Kn is set to 0.01 and 0.05. Besides, a continuum flow (Kn → 0) at Re = 10 is also presented for comparison. For the stress force acting on the surface of the cylinder, it can be obtained from the distribution function [49] in a rarefied flow regime, since the Newtonian friction law fails in that flow regime. In the continuum regime, $C_l'$ continues to decline with the increase in the $U^*$ value. This is different in the case of Re = 33, where the minimum $C_l'$ is reached at $U^* = 10$. The possible reason is that the influence of vortex separation on the result is decreased at a low Reynolds number. Furthermore, as shown in Fig. 25(a), this phenomenon can also be observed in a rarefied flow regime, where $C_l'$ is shown in logarithmic coordinates. Due to the rarefied gas effect, the no-slip wall boundary condition cannot be maintained, and the lift and drag forces decrease. In addition, for drag force on a circular



(a) (b)

FIG. 26. Convergence tests for the lift coefficient $C_l$ and the drag coefficient $C_d$ at (a) Kn = 0.01 and (b) Kn = 0.05. Single $f$: only $g$ is solved. Double $f$: both $g$ and $h$ are solved. 16GH: 16-abscissas Gauss–Hermit quadrature rule.

cylinder at different reduced frequencies, a linear relation similar to that of lift force can also be observed in the rarefied flow regime [see Fig. 25(b)]. Consequently, for rarefied flow at these two Kn numbers, the lift and drag forces on the oscillating circular cylinder exhibit a linear relation with the Knudsen number Kn, Reynolds number Re, amplitude of oscillation $A$, and oscillating frequency $1/U^*$. Further studies at other computational conditions will be continued and will be presented in our future papers.

Finally, we discuss the convergence test case, where the Gauss–Hermit quadrature rule is used to integrate the macro-quantities, and $U^*$ is set to 5. As shown in Fig. 26(a), for $\mathrm{Kn} = 0.01$, because the Mach number Ma of flow is low ($\mathrm{Ma} \approx 0.06$), only the density distribution function $g$ is solved, and 16 abscissas sufficiently provide convergent results. For $\mathrm{Kn} = 0.05$ ($\mathrm{Ma} \approx 0.3$), although the differences are little [see Fig. 26(b)], both the density distribution function $g$ and the energy distribution function $h$ are solved, and 28 abscissas are used in the simulation to provide convergent results.

## IV. CONCLUSION

In the present study, the original DUGKS is extend to an ALE-type DUGKS. The mesh motion velocity is introduced in the Boltzmann–BGK equation to modify the net flux of a cell interface. Consequently, based on the constructed mesh motion velocity, the remapping-free-type ALE method is used to develop the current ALE-type DUGKS. To exclude the GCL error, three DGCL-compliant schemes are discussed. As for the proposed DUGKS, only the intermediate time level is needed to calculate the flux of a cell interface, based on the geometry average, the geometrical information at this time level is easy to define, which indicates DGCL scheme 1 is the most suitable. Further improved work such as a high-order or multi-time-level implicit ALE-type DUGKS, where defining the geometrical information at multi-time-level is not easy, DGCL scheme 2 or scheme 3 is the most suitable. A uniform flow test case indicates that all the three schemes exhibit good performances since no disturbances are introduced in the computational domain. Five test cases of low-speed flow are simulated, namely, two continuum flow conditions and three rarefied flow conditions. The results of all the test cases agree well with existing numerical and experimental results. Therefore, for continuum flows, similar to the macromethods based on the N–S equations, the proposed ALE-type DUGKS has the capability to simulate more complex low-speed moving boundary problems. Further, rarefied flows, which are out of the computational range for macromethods, can also be handled by the proposed method. Further works such as parallel computing and implicit accelerated methods will be continued to enhance the ability of the proposed ALE-type DUGKS to simulate the moving boundary problems at different flow regimes.

## ACKNOWLEDGMENTS

This study is supported by the National Natural Science Foundation of China (Grant No. 11472219) and the 111 Project of China (Grant No. B17037).

[1] L. Zhang, X. Chang, X. Duan, Z. Zhao, and X. He, Comput. Fluids **62**, 45 (2012).

[2] J. Whitney and R. Wood, Bioinspir. Biomimet. **7**, 036001 (2012).

[3] H.-S. Tsien, J. Aeronaut. Sci. **13**, 653 (1946).

[4] T. Tsuji and K. Aoki, J. Comput. Phys. **250**, 574 (2013).

[5] W. Shyy, H. Udaykumar, M. Rao, and R. Smith, AIAA J. **36**, 303 (1998).

[6] S. Shrestha, S. Tiwari, A. Klar, and S. Hardt, J. Comput. Phys. **292**, 239 (2015).

[7] A. Patronis, D. A. Lockerby, M. K. Borg, and J. M. Reese, J. Comput. Phys. **255**, 558 (2013).

[8] Z. Guo, K. Xu, and R. Wang, Phys. Rev. E **88**, 033305 (2013).

[9] S. Succi, *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond* (Oxford University Press, Oxford, 2001).

[10] K. Xu and J.-C. Huang, J. Comput. Phys. **229**, 7747 (2010).

[11] Z. Guo, R. Wang, and K. Xu, Phys. Rev. E **91**, 033313 (2015).

[12] P. Wang, L. Zhu, Z. Guo, and K. Xu, Commun. Comput. Phys. **17**, 657 (2015).

[13] L. Zhu, Z. Guo, and K. Xu, Comput. Fluids **127**, 211 (2016).

[14] D. Pan, C. Zhong, and C. Zhuo, Commun. Comput. Phys. **25**, 1469 (2019).

[15] R. Mittal and G. Iaccarino, Annu. Rev. Fluid Mech. **37**, 239 (2005).

[16] R. Yuan, C. Zhong, and H. Zhang, J. Comput. Phys. **296**, 184 (2015).

[17] R. Yuan and C. Zhong, Appl. Math. Model. **55**, 417 (2018).

[18] L. K. Ragta, B. Srinivasan, and S. S. Sinha, Int. J. Numer. Methods Fluids **85**, 507 (2017).

[19] J. M. Stockie, J. A. Mackenzie, and R. D. Russell, SIAM J. Sci. Comput. **22**, 1791 (2001).

[20] W.-H. Hui and K. Xu, *Computational Fluid Dynamics Based on the Unified Coordinates* (Springer, Berlin, 2012).

[21] C. W. Hirt, A. A. Amsden, and J. Cook, J. Comput. Phys. **14**, 227 (1974).

[22] X. Chang, R. Ma, L. Zhang, X. He, and M. Li, Comput. Fluids **120**, 98 (2015).

[23] J. Yang and J. Huang, J. Comput. Phys. **120**, 323 (1995).

[24] Y.-H. Qian, D. d'Humières, and P. Lallemand, Europhys. Lett. **17**, 479 (1992).

[25] S. Chen, K. Xu, C. Lee, and Q. Cai, J. Comput. Phys. **231**, 6643 (2012).

[26] C. Jin and K. Xu, J. Comput. Phys. **222**, 155 (2007).

[27] R. Löhner and C. Yang, Commun. Numer. Methods Eng. **12**, 599 (1996).

[28] P. Thomas and C. Lombard, in *Proceedings of the 11th Fluid and PlasmaDynamics Conference* (AIAA, 1978), p. 1208.

[29] M. Lesoinne and C. Farhat, Comput. Methods Appl. Mech. Eng. **134**, 71 (1996).

[30] C. Wu, B. Shi, C. Shu, and Z. Chen, Phys. Rev. E **97**, 023306 (2018).

[31] B. Koobus and C. Farhat, Comput. Methods Appl. Mech. Eng. **170**, 103 (1999).

[32] Z. Guo, C. Zheng, and B. Shi, Phys. Fluids **14**, 2007 (2002).

[33] T. Tsuji and K. Aoki, Microfluid. Nanofluid. **16**, 1033 (2014).

[34] F. Archambeau, N. Méchitoua, and M. Sakiz, Int. J. Finite Vol. **1**, 1 (2004).

[35] H. Guillard and C. Farhat, Comput. Methods Appl. Mech. Eng. **190**, 1467 (2000).

[36] H. Dütsch, F. Durst, S. Becker, and H. Lienhart, J. Fluid Mech. **360**, 249 (1998).

[37] Z. Guo, B. Shi, and N. Wang, J. Comput. Phys. **165**, 288 (2000).

[38] J. Morison, J. Johnson, S. Schaaf *et al.*, J. Pet. Technol. **2**, 149 (1950).

[39] B. Uzunoğlu, M. Tan, and W. Price, Int. J. Numer. Methods Eng. **50**, 2317 (2001).

[40] J. Young and J. C. S. Lai, AIAA J. **42**, 2042 (2004).

[41] C. Liang, K. Ou, S. Premasuthan, A. Jameson, and Z. Wang, Comput. Fluids **40**, 236 (2011).

[42] M. M. Koochesfahani, AIAA J. **27**, 1200 (1989).

[43] R. Ramamurti and W. Sandberg, AIAA J. **39**, 253 (2001).

[44] A. Mackowski and C. Williamson, J. Fluid Mech. **765**, 524 (2015).

[45] G. Dechristé and L. Mieussens, in *Journal of Physics: Conference Series* (IOP Publishing, Bristol, UK, 2012), Vol. 362, p. 012030.

[46] W. H. Press, *Numerical Recipes in Fortran 77: The Art of Scientific Computing* (Cambridge University Press, Cambridge, 1992).

[47] K. Aoki, S. Kosuge, T. Fujiwara, and T. Goudon, Phys. Rev. Fluids **2**, 013402 (2017).

[48] S. Mittal and S. Singh, J. Fluid Mech. **534**, 185 (2005).

[49] K. Xu, *Direct Modeling for Computational Fluid Dynamics: Construction and Application of Unified Gas-Kinetic Schemes* (World Scientific, Singapore, 2015).